



**INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA**

Exercices et solutions 2016 Tout Age

<http://www.castor-informatique.ch/>

Éditeurs :

Julien Ragot, Gabriel Parriaux, Jean-Philippe Pellet, Nicole Müller, Christian Datzko, Hanspeter Erni

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erusbildung//sociétésuisse del'inform
atique dans l'enseignement//societàsviz
zera perl'informaticanell'insegnamento



Ont collaboré au Castor Informatique 2016

Andrea Adamoli, Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Martin Guggisberg, Corinne Huck, Carla Monaco, Nicole Müller, Gabriel Parriaux, Jean-Philippe Pellet, Julien Ragot, Beat Trachsler.

Nous adressons nos remerciements à :

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Tobias Kohn, Ivana Kosírová, Serena Pedrocchi, Björn Steffen : ETHZ

Valentina Dagiene : Bebras.org

Hans-Werner Hein, Wolfgang Pohl, Peter Rossmann : Bundesweite Informatikwettbewerbe (BWINF), Allemagne

Anna Morpurgo, Violetta Lonati, Mattia Monga : Italie

Gerald Futschek : Austrian Computer Society, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Eljakim Schrijvers, Daphne Blokhuis, Marissa Engels : Eljakim Information Technology by, Pays-Bas

Roman Hartmann : hartmannGestaltung (Flyer Castor Informatique Suisse)

Christoph Frei : Chragokyberneticks (Logo Castor Informatique Suisse)

Pamela Aeschlimann, Andreas Hieber, Aram Loosmann, Daniel Vuille, Peter Zurflüh : Lernetz.ch (page web)

Andrea Leu, Maggie Winter, Brigitte Maurer : Senarclens Leu + Partner

La version allemande des exercices a également été utilisée en Allemagne et en Autriche.

L'adaptation française a été réalisée par Nicole Müller et la version italienne par Andrea Adamoli.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Le Castor Informatique 2016 a été réalisé par la Société Suisse de l'Informatique dans l'Enseignement SSIE. Le Castor Informatique est un projet de la SSIE, aimablement soutenu par la Fondation Hasler.

HASLERSTIFTUNG

Tout lien a été vérifié le 1 novembre 2016. Ce cahier d'exercice a été produit le 13 novembre 2016 avec avec le logiciel de mise en page L^AT_EX.



Les exercices sont protégés par une licence Creative Commons Paternité – Pas d'Utilisation Commerciale – Partage dans les Mêmes Conditions 4.0 International. Les auteurs sont cités p. 103.



Préambule

Très bien établi dans différents pays européens depuis plusieurs années, le concours « Castor Informatique » a pour but d'éveiller l'intérêt des enfants et des jeunes pour l'informatique. En Suisse, le concours est organisé en allemand, en français et en italien par la SSIE, la Société Suisse de l'Informatique dans l'Enseignement, et soutenu par la Fondation Hasler dans le cadre du programme d'encouragement «FIT in IT».

Le Castor Informatique est le partenaire suisse du concours «Bebras International Contest on Informatics and Computer Fluency» (<http://www.bebbras.org/>), initié en Lituanie.

Le concours a été organisé pour la première fois en Suisse en 2010. Le Petit Castor (années scolaire 3 et 4) a été organisé pour la première fois en 2012.

Le Castor Informatique vise à motiver les élèves pour l'informatique. Il souhaite lever les réticences et susciter l'intérêt quant à l'enseignement de l'informatique à l'école. Le concours ne suppose aucun prérequis dans l'utilisation des ordinateurs, sauf savoir «surfer» sur Internet, car le concours s'effectue en ligne sur un PC. Pour répondre, il faut structurer sa pensée, faire preuve de logique mais aussi de fantaisie. Les exercices sont expressément conçus pour développer un intérêt durable pour l'informatique, au-delà de la durée du concours.

Le concours Castor Informatique 2016 a été fait pour cinq tranches d'âge, basées sur les années scolaires :

- Années scolaires 3 et 4 (Petit Castor)
- Années scolaires 5 et 6
- Années scolaires 7 et 8
- Années scolaires 9 et 10
- Années scolaires 11 à 13

Les élèves des années scolaires 3 et 4 avaient 9 exercices à résoudre (3 faciles, 3 moyens, 3 difficiles). Chaque autre tranche d'âge devait résoudre 15 exercices, dont 5 de degré de difficulté facile, 5 de degré moyen et 5 de degré difficile.

Chaque réponse correcte donnait des points, chaque réponse fautive réduisait le total des points. Ne pas répondre à une question n'avait aucune incidence sur le nombre de points. Le nombre de points de chaque exercice était fixé en fonction du degré de difficulté :

	Facile	Moyen	Difficile
Réponse correcte	6 points	9 points	12 points
Réponse fautive	-2 points	-3 points	-4 points

Utilisé au niveau international, ce système de distribution des points est conçu pour limiter le succès en cas de réponses données au hasard.

Les participants disposaient de 45 points (Petit Castor 27) sur leur compte au début du concours. Le maximum de points possibles était de 180 points (Petit Castor 108), le minimum étant de 0 point. Les réponses de nombreux exercices étaient affichées dans un ordre établi au hasard. Certains exercices ont été traités par plusieurs tranches d'âge.

Pour de plus amples informations :

SVIA-SSIE-SSII (Société Suisse de l'Informatique dans l'Enseignement)

Castor Informatique

Julien Ragot

castor@castor-informatique.ch

<http://www.castor-informatique.ch/>



 <https://www.facebook.com/informatikbiberch>



Table des matières

Ont collaboré au Castor Informatique 2016	i
Préambule	ii
1. Les boules de glace	1
2. Irrigation	3
3. Une soirée karaoké	5
4. À qui la faute ?	7
5. Que le message passe !	9
6. Paires correspondantes	11
7. La souris et le fromage	13
8. Courriel du casino	17
9. Une étagère bien rangée	19
10. Une recette secrète	23
11. Que les fleurs fleurissent !	25
12. Un anniversaire à deux valeurs	27
13. Tous en même temps	31
14. Le parcours de la boule	35
15. Des fleurs et des soleils	37
16. Le tchat des castors	39
17. Quatre courses à faire	41
18. Des impasses	43
19. Des messages secrets	45
20. Vive la hiérarchie !	47
21. Plongeons-nous dans la construction des ponts !	49
22. Des cônes servant de cachette	51
23. Le travail en groupe	53
24. Des bonbonnières	57



25. 60 minutes. . . Zut !	59
26. Le code KIX	61
27. Le filtre médian	63
28. La poignée de main style zip	65
29. Les jeux de Nim	69
30. Le classement des numéros de maillot	71
31. Cassy, la tortue	73
32. Chaque palette en son temps !	77
33. Jeux de boule	81
34. Deux possibilités exclusives	85
35. Le carrefour des trams	87
36. Le codage des images numériques	89
37. Las de nouer !	93
38. Peinture récursive	95
39. Le jeu de formes géométriques	97
40. Les écureuils égoïstes	101
A. Auteurs des exercices	103
B. Sponsoring : Concours 2016	104
C. Offres ultérieures	106



1. Les boules de glace

Chez le glacier LIFO, on empile les boules de glace aux arômes désirés dans un cornet à glace et ceci, précisément dans l'ordre demandé par le client.

Que doit dire le client s'il aimerait goûter une glace comme le montre la figure de droite ?

J'aimerais une glace ...

- A) ... au chocolat, à la menthe et à la myrtille!
- B) ... au chocolat, à la myrtille et à la menthe!
- C) ... à la myrtille, à la menthe et au chocolat!
- D) ... à la myrtille, au chocolat et à la menthe!





Solution

La réponse correcte est C) : «J'aimerais une glace à la myrtille, à la menthe et au chocolat.»
Ce qui est mentionné en premier est mis tout en bas de la pile. Ce qui est mentionné en dernier est mis tout en haut de la pile. Dans la réponse A), l'ordre est carrément inversé. Dans la réponse B) et D), la glace à la menthe ne se trouve pas au milieu de la pile.

C'est de l'informatique !

Il est important savoir respecter l'ordre. Si le client indique les arômes de glace dans un ordre différent, l'ordre de la pile des glaces changera également.

En informatique, on apprend combien il est important de classer les choses ainsi que de savoir appliquer un ordre adéquat à la situation en question. Il s'ajoute qu'il est indispensable de connaître le comportement du glacier ainsi que d'analyser à fond le problème qu'il se pose. Sans avoir compris la situation en cause, aucun informaticien n'est capable de développer un programme adéquat.

L'ordre utilisé dans cette tâche est appelé LIFO, «last in, first out», ce qui signifie en français : «dernier arrivé, premier sorti».

Sites web et mots clés

Last In, First Out, souvent abrégé par l'acronyme LIFO, signifie « dernier arrivé, premier sorti », Pile, Structures de données, Last In, First Out, souvent abrégé par l'acronyme LIFO, signifie « dernier arrivé, premier sorti »

— http://fr.wikipedia.org/wiki/Last_in,_first_out



2. Irrigation

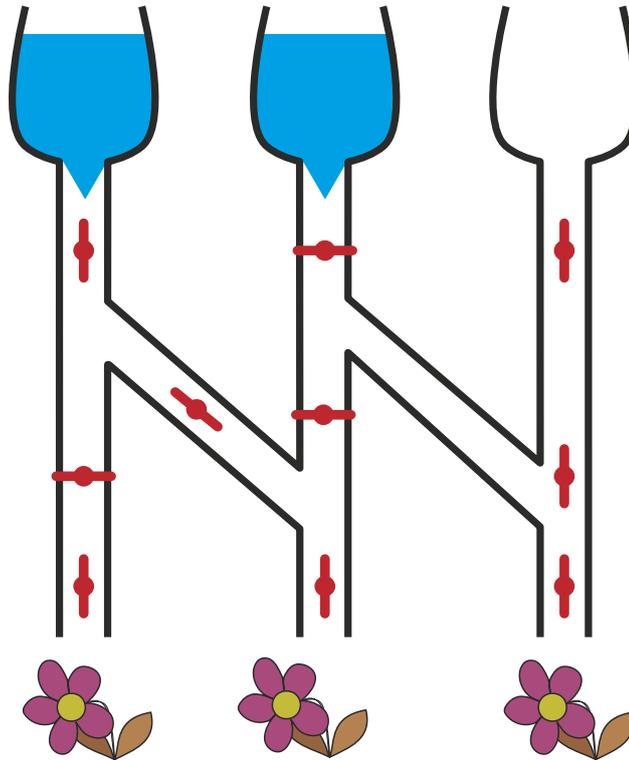
Lorsque la valve est ouverte, l'eau s'écoule :



Lorsque la valve est fermée, l'eau ne s'écoule pas :



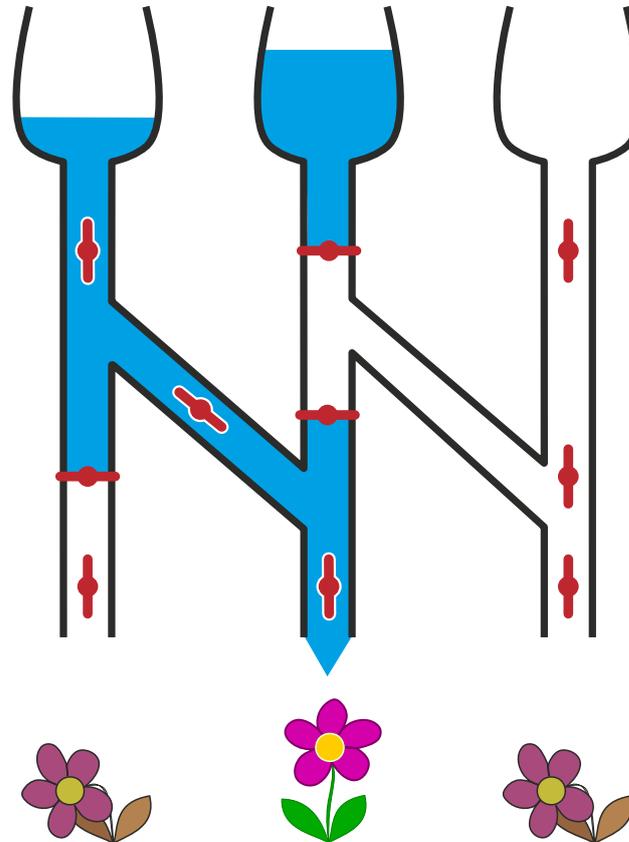
Lesquelles de ces trois fleurs seront arrosées lorsque les valves sont ainsi positionnées ?





Solution

Seule la fleur du milieu sera arrosée avec cette position de la valve.



C'est de l'informatique !

Pour l'informatique, notre système d'irrigation est un circuit. Les soupapes sont les interrupteurs – avec les deux positions « marche » et « arrêt ». En fonction des trémies d'alimentation et des positions des interrupteurs, les informations « L'eau s'écoule » et « L'eau ne s'écoule pas » se déplacent à travers le circuit, jusqu'aux fleurs.

Les appareils électroniques contiennent des circuits électroniques à travers lesquels circule l'électricité. Dans les circuits en fibres de verre, les informations circulent sous la forme d'une lumière laser.

Il existe des robots qui doivent travailler dans des environnements dans lesquels les circuits électroniques tombent rapidement en panne : champs magnétiques puissants, humidité élevée, températures extrêmes. Cet équipement robotique doit contenir des circuits très solides dans lesquels circulent de l'huile hydraulique ou de l'air comprimé.

Sites web et mots clés

Circuits

— https://fr.wikipedia.org/wiki/Circuit_électronique



3. Une soirée karaoké

Quelques castors aimeraient bien participer à une soirée karaoké ensemble. Pour faire jouer de la musique karaoké, ils ont recours à quatre programmes différents. Chaque programme propose un réglage de volume sonore différent.

Choisis le programme où le volume sonore est réglé le plus fort.





Solution

La réponse C) est correcte :

- Dans le programme «Karaoke Sound 3.0», le volume sonore est réglé à mi-hauteur.
- Dans le programme «AudaBeaverCity», le volume sonore est réglé très bas.
- Dans le programme «Lake Sound Master», le volume sonore est réglé le plus haut, il est à plein volume.
- Dans le programme «Beaver Karaoke Player 7», la musique est réduite au silence, donc, on n'entend rien du tout.

C'est de l'informatique !

Bien que tu n'aies jamais vu un de ces programmes, tu as été tout à fait capable de reconnaître comment le réglage du volume fonctionne pour chaque programme. Ceci est dû au fait que beaucoup de programmes utilisent les mêmes symboles.

Dans notre cas, le symbole d'un haut-parleur représente le réglage du volume. En effet, tous les quatre programmes utilisent ce même symbole. Bien que son aspect change, le concept est le même. Le volume réglé est représenté soit par une certaine quantité de barres soit par une ligne toujours plus épaisse. Si tu aperçois un haut-parleur barré ou superposé d'une croix, tu comprends tout de suite que ce dernier est éteint.

Des programmes utiles et faciles à comprendre se démarquent par le respect de tels symboles typiques et intelligibles par tout le monde pour déterminer des fonctions. Par ailleurs, il est important que l'utilisateur retrouve les symboles dont il a besoin à des endroits où il les attend. C'est ainsi que l'utilisateur se retrouvera rapidement dans un nouveau système sans qu'il soit obligé de consulter de gros manuels.

Sites web et mots clés

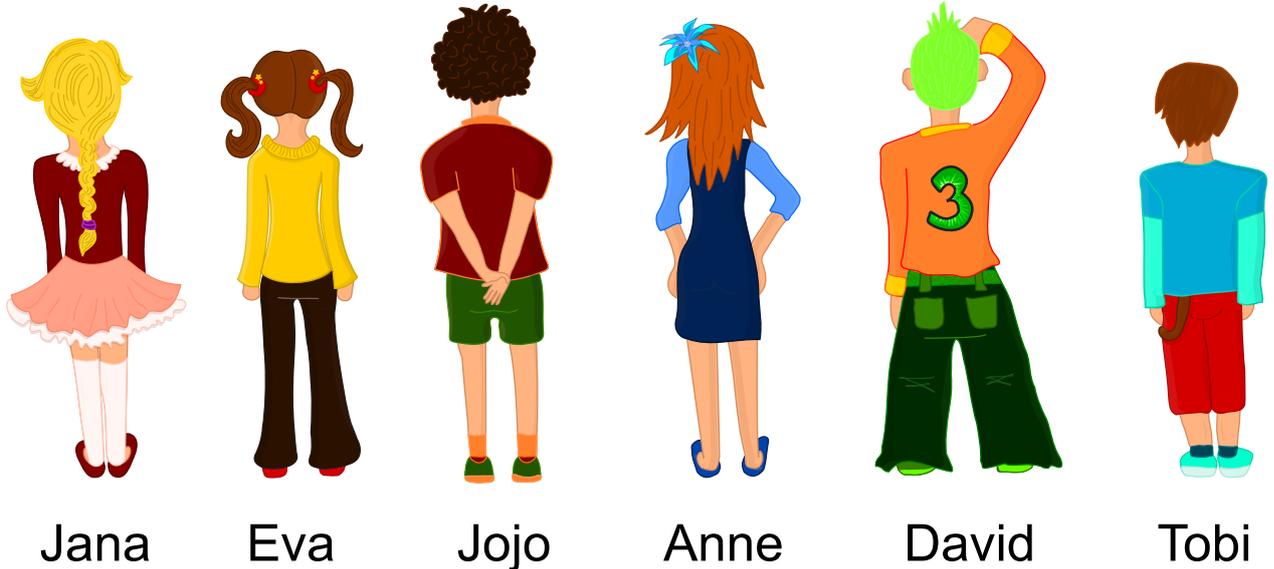
conception d'interfaces graphiques («GUI design»), interface utilisateur

- https://en.wikipedia.org/wiki/User_interface_design



4. À qui la faute ?

Six enfants sont en train de jouer au ballon en plein air.



Tout à coup, j'entends le bruit de bris de verre ! Je me précipite à la fenêtre et, à travers le verre brisé, j'aperçois un enfant partir en courant. L'enfant a les cheveux longs et porte une paire de pantalons longs.

Qui parmi les six enfants a brisé la vitre de fenêtre ?

- A) Jana
- B) Eva
- C) Jojo
- D) Anne
- E) David
- F) Tobi



Solution

La réponse correcte est B). C'est Eva.

Parmi les six enfants, il n'y a que trois enfants qui ont les cheveux longs : Jana, Eva et Anne. En plus, il n'y a que deux enfants qui portent des pantalons longs : Eva et David. Comme Eva est la seule qui correspond aux deux caractéristiques recherchées, ce doit être elle qui a brisé la vitre de la fenêtre.

C'est de l'informatique !

Pour les modèles de données, l'informatique regroupe souvent des objets selon leurs caractéristiques (le terme mieux approprié est : «propriété» ou «attribut»). Dans la tâche présente, nous devons tenir compte de deux caractéristiques et de six personnes.

Nous cherchons la personne qui répond aux deux caractéristiques. Quand on parle de grands ensembles de données, une telle opération est aussi appelée : «calculer l'intersection de deux ensembles». C'est ainsi que l'on arrive à identifier («filtrer») des objets à l'aide des propriétés déterminées et ceci, même à l'intérieur des quantités de données énormes («construire un sous-ensemble»).

Sites web et mots clés

attribut, critères de sélection, base de données

— [https://fr.wikipedia.org/wiki/Objet_\(informatique\)](https://fr.wikipedia.org/wiki/Objet_(informatique))

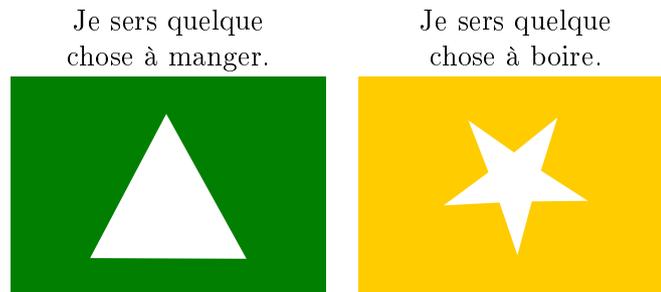


5. Que le message passe !

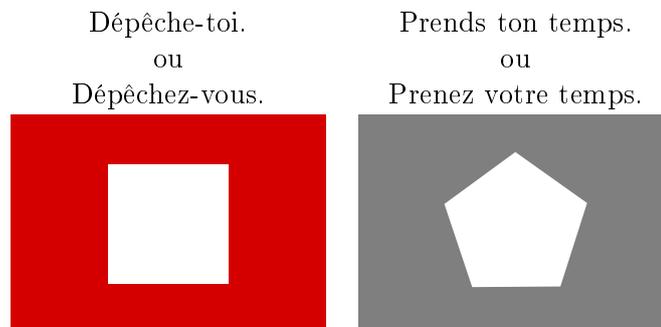
Le père de Béatrice et de Richard, Albert, est maître-nageur sur une plage. Béatrice et Richard sont en train de jouer avec les autres enfants sur la plage. Afin de pouvoir communiquer avec Béatrice et Richard, Albert prend recours à un poteau et des drapeaux comportant de différents codes. Le drapeau placé tout en haut peut avoir les significations suivantes :



Le drapeau placé au milieu peut avoir les significations suivantes :

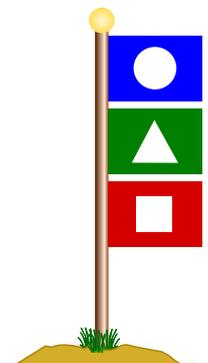


Le drapeau placé tout en bas peut avoir les significations suivantes :



Albert hisse trois drapeaux sur le poteau. Choisis le message qu'il aimerait bien transmettre à ses enfants.

- A) Béatrice, je sers quelque chose à manger, dépêche-toi.
- B) Richard, je sers quelque chose à manger, dépêche-toi.
- C) Richard et Béatrice, je sers quelque chose à boire, prenez votre temps.
- D) Béatrice, je sers quelque chose à boire, prends ton temps.





Solution

La réponse B) est correcte : Le drapeau bleu signifie que le message s'adresse à Richard. Le drapeau vert signifie qu'Albert sert quelque chose à manger. Le drapeau rouge veut dire que la personne à laquelle le message s'adresse doit se dépêcher.

C'est de l'informatique !

Si Albert utilisait un seul drapeau pour chacun des messages, le décodage du message serait ultra facile. Afin de décoder le message correct de l'exercice présent, il faut tenir compte non seulement du code particulier de chacun de ces drapeaux mais aussi de la disposition de ces-derniers sur le poteau : leur placement, en haut, au milieu ou en bas, nous renseigne sur la fonction qu'ils accomplissent dans l'ensemble du message.

Le concept du '«if» (condition) ... «then» (code exécuté si la condition est vérifiée)' est très connu dans le domaine de l'informatique. Prenons par exemple un smartphone : le bouton «home» comporte plusieurs significations en fonction de la situation. Ou bien si, par exemple, on est en train de jouer à un jeu vidéo sur ordinateur et que l'on se sert du bouton «home» sur le clavier, la fonction de la commande sera une autre que si on se sert du même bouton en travaillant avec un programme de traitement de texte ou avec un programme d'édition d'image.

Dans le domaine de la programmation, on connaît également les blocs '«if» ... «when»'. En fonction de différentes entrées, le programme exécute différentes opérations. Ainsi, en analogie avec le concept '«if» ... «then»', le comportement de Béatrice et Richard dépend aussi des différentes dispositions des drapeaux sur le poteau.

Sites web et mots clés

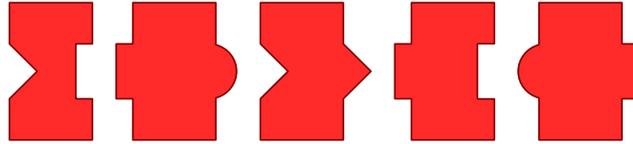
code , sélection, programmer

- <https://fr.wikipedia.org/wiki/Branchement>
- [https://fr.wikipedia.org/wiki/Code_\(information\)](https://fr.wikipedia.org/wiki/Code_(information))



6. Paires correspondantes

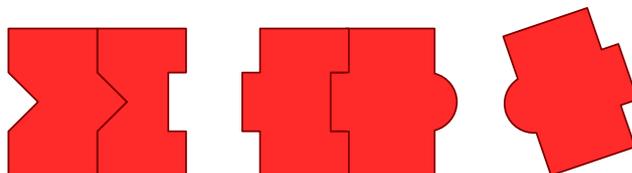
Les castors se réjouissent d'un nouveau jeu de puzzle qui comprend cinq pièces de puzzle. Quelques pièces peuvent être imbriquées à l'aide de connecteurs adéquats afin de les assembler en paires. *Assemble autant de paires que possible. Imbrique les pièces de puzzle à l'aide de la souris. Note que l'assemblage ne fonctionne que si les connecteurs ont la forme adéquate.*





Solution

Il est possible d'assembler les deux paires suivantes :



Au cas où on imbrique deux autres pièces, on ne pourra créer qu'une seule paire. De toutes les pièces, il en restera donc trois pièces que l'on n'arrivera pas à assembler.

C'est de l'informatique !

Comment est-ce que tu as résolu ce problème ? Probablement, tu as simplement essayé d'assembler des pièces possibles pour conclure, que finalement, tu ne pouvais créer qu'une seule paire.

Pour des tâches plus complexes, l'ordinateur, lui aussi, peut être utilisé pour tester toutes les options possibles afin de parvenir à la solution souhaitée. Tant que la tâche n'est pas trop complexe, ce procédé est tout à fait valable. Mais quand il s'agit d'un problème à caractère plus vaste, il se peut que même un ordinateur ait besoin de plusieurs années pour le résoudre. Dans ce cas-là, on a plutôt recours à une recherche «heuristique». Avec ce procédé, on cherche à parvenir à la solution de manière optimisée : ainsi, on évite que l'ordinateur doive examiner toutes les options possibles pour arriver à la solution souhaitée et ceci, à l'aide de règles simples pour décider au cours du processus quelle approche sera favorisée et laquelle, par contre, sera abandonnée.

Sites web et mots clés

recherche par force brute, heuristique, algorithme glouton

- https://fr.wikipedia.org/wiki/Recherche_exhaustive
- [https://fr.wikipedia.org/wiki/Heuristique_\(mathématiques\)](https://fr.wikipedia.org/wiki/Heuristique_(mathématiques))
- https://fr.wikipedia.org/wiki/Algorithme_glouton



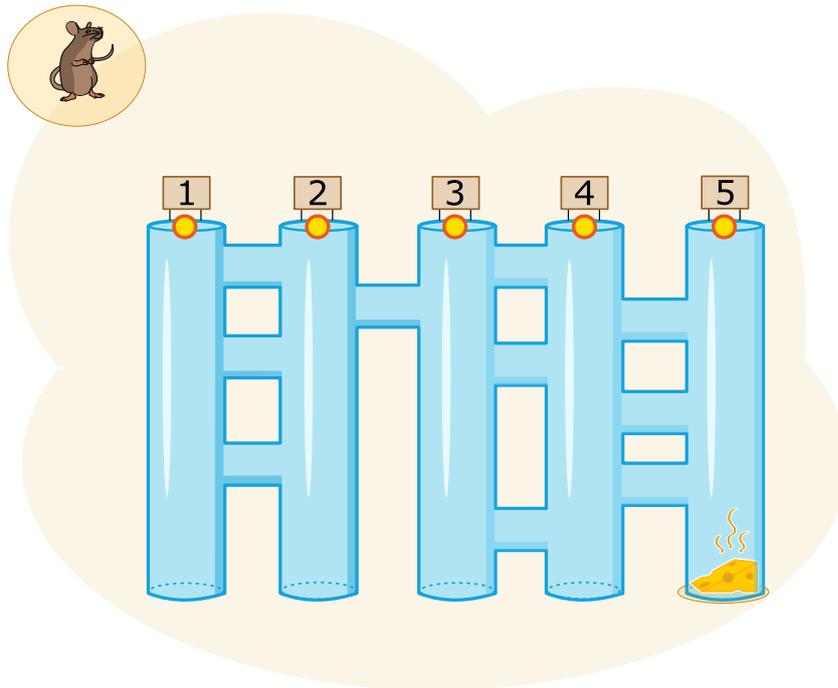
7. La souris et le fromage

Une souris se trouve tout en haut à l'entrée de cinq longs couloirs. Ces cinq longs couloirs sont reliés par des passages transversaux courts. Alléchée par le fromage qui se trouve en bas à la fin du long couloir à l'extrême droite, la souris aimerait bien parcourir ce labyrinthe le plus vite possible.

Sur son chemin, elle suit les commandes suivantes, et ceci toujours en alternance :

- Parcours le long couloir vers le bas jusqu'à ce que tu arrives à un passage transversal.
- Parcours le passage transversal.

Lequel des cinq longs couloirs doit-elle choisir pour arriver au fromage ?





Malheureusement, en exécutant les instructions, l'ordinateur, cette machine logique, n'est pas capable de réfléchir de manière logique. Si la souris de notre exercice était un être vivant et qu'elle avait choisi le couloir n° 1, elle aurait senti et vu le fromage déjà au bout du couloir n° 5. Ainsi, elle n'aurait donc jamais parcouru le passage transversal du couloir n° 4 pour arriver au bout du couloir n° 4 où il n'y a pas de fromage. La souris de notre exercice, par contre, ne fait que suivre de manière précise les instructions qui lui ont été dictées (tout comme un ordinateur). C'est la raison pour laquelle elle risque de suivre un chemin erroné.

Ce qui est important pour toi : si tu utilises un ordinateur, sois certain qu'il exécutera exactement ce que tu lui as dicté préalablement ... et non pas ce que tu désires ou t'imagines qu'il exécutera.

Sites web et mots clés

ordinateur, programme

— https://fr.wikipedia.org/wiki/Programme_informatique





8. Courriel du casino

Julie reçoit cette publicité dans un courriel :



Elle veut absolument faire partie des gagnants! À l'école, elle a entendu dire que seuls les adultes ont le droit de participer à des jeux de chance parce que les chances de gagner sont minimes et qu'en plus, les données personnelles des joueurs peuvent être détournées à des fins malhonnêtes.

Que doit-elle faire ?

- A) Faire comme si elle était sa mère et participer avec les données personnelles de celle-ci.
- B) Participer tout simplement avec ses propres données.
- C) Supprimer le message.
- D) Demander à son frère plus âgé de participer avec ses données personnelles.



Solution

La réponse C est correcte : Julie devrait tout de suite supprimer ce message, car il s'agit d'un spam, aussi appelé pourriel. Dans le cas des réponses A, B et D elle devrait envoyer des données personnelles, les siennes ou celles d'autres personnes de sa famille. Ces données pourraient être utilisées à des fins illicites. Par exemple, pour envoyer de façon ciblée encore plus de pourriels. Mais il pourrait aussi s'agir de choses plus graves.

C'est de l'informatique !

Un spam coûte cher. Le temps que des millions d'utilisateurs perdent quotidiennement à lire et à effacer des spams représente un dommage économique considérable.

Le spam est un média de masse utilisé pour la diffusion de logiciels malveillants. Nombreux sont les utilisateurs qui ne remarquent pas que leurs ordinateurs sont d'ores et déjà commandés à distance à leur insu pour envoyer des spams. Il suffit pour cela d'avoir cliqué sur un spam alors que votre logiciel de protection n'était pas à jour sur votre ordinateur et le mal est fait.

L'informatique s'efforce d'améliorer non seulement les filtres anti-spams automatiques et d'autres logiciels de protection, mais aussi d'informer la population des risques et des obligations de vigilance liés à leur participation à Internet.

Les jeunes devraient notamment être instruits suffisamment tôt sur les raisons justifiant qu'il leur est interdit de participer à des jeux de chance sur Internet. Pourquoi? Parce que c'est stupide. Il est impossible d'avoir confiance dans le véritable hasard du jeu et en plus notre sécurité est mise en danger.

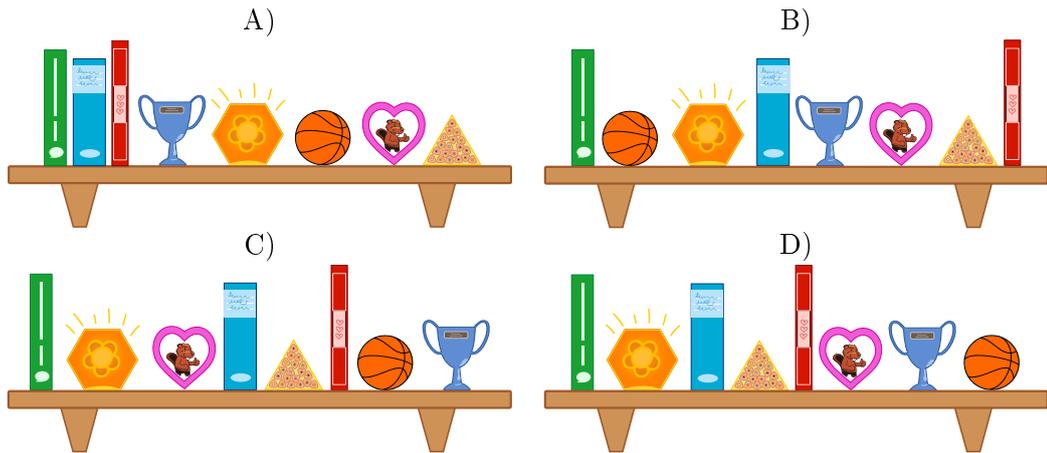


9. Une étagère bien rangée

Béatrice réorganise son étagère. Elle décide de classer les objets selon deux conditions :

1. Les objets rectangulaires *ne doivent* pas être placés les uns à côté des autres.
2. Les objets ronds *ne doivent* pas être placés à côté des objets rectangulaires.

Quelle planche d'étagère est arrangée selon les deux conditions imposées par Béatrice ?





Solution

La réponse D) est correcte :



Le rangement des objets de la planche d'étagère A) ne répond pas à la première condition parce que les objets rectangulaires de l'extrême gauche sont placés les uns à côté des autres.

La planche d'étagère B) n'est pas organisée de manière correcte car le ballon, un objet rond, est placé à côté d'un objet rectangulaire.

La planche d'étagère C) n'est pas organisée de manière correcte car le ballon, un objet rond, est placé à côté d'un objet rectangulaire.

Le rangement des objets de la planche d'étagère D), par contre, répond aux deux conditions imposées par Béatrice.

C'est de l'informatique !

Ne nous y trompons pas : les étagères ne sont pas toujours joliment rangées. Dans le domaine de l'informatique, l'organisation des fichiers peut également poser certains problèmes. Si, par exemple, on écrit, supprime ou déplace des fichiers sur le disque dur d'un ordinateur, les fichiers d'origine demeurent à leur emplacement d'origine. Ceci peut provoquer un désordre dans le stockage des fichiers. Dans le domaine de l'informatique, un tel désordre est appelé «fragmentation».

Comme le disque dur d'un ordinateur est divisé en secteurs et que le système d'exploitation les rassemble pour former des blocs de données, le contenu d'un fichier est stocké sur un certain nombre de blocs. Or, rien ne garantit que le système de fichiers utilise des blocs physiquement voisins. Il se peut donc que la tête de lecture doive aller chercher l'ensemble des blocs éparpillés sur la surface du disque dur. Ce processus nécessite beaucoup de temps : la lecture du même nombre de blocs ne s'effectuera par exemple qu'avec 10 mégaoctets au lieu de 100 mégaoctets par seconde. Le disque dur est donc «fragmenté».

Pour remédier à la fragmentation, on pourrait veiller à classer et à stocker les blocs des fichiers dès le début de manière judicieuse : on pourrait par exemple sauvegarder les blocs avec peu de données sur des espaces restreints et les blocs avec beaucoup de données, sur des espaces plus spacieux. Ou bien... et encore... mais l'examen de toutes ces possibilités prend trop de temps... Habituellement, on a tout simplement recours à un processus de réorganisation du disque dur déjà existant, à savoir au processus de la «défragmentation».

Pourtant, le problème de la fragmentation ne se pose pas pour tous les supports de stockage. Pendant que les disques durs et surtout les bandes magnétiques en sont affectés, les disques durs SSD (Solid-state drive) ainsi que les clés USB (Bus universel en Série) ne le sont pas. C'est parce que l'accès aux données des fichiers ne fonctionne pas à travers le déplacement des fragments. Bien au contraire : le processus de la défragmentation signifie d'exécuter des opérations d'écriture illimitée, et les supports SSD ou USB ne permettent que des cycles d'écriture limités.



Reste à savoir si le classement de Béatrice est pertinent... c'est à chacun de choisir !

Sites web et mots clés

données, règles, caractéristiques de forme

— [https://fr.wikipedia.org/wiki/Défragmentation_\(informatique\)](https://fr.wikipedia.org/wiki/Défragmentation_(informatique))

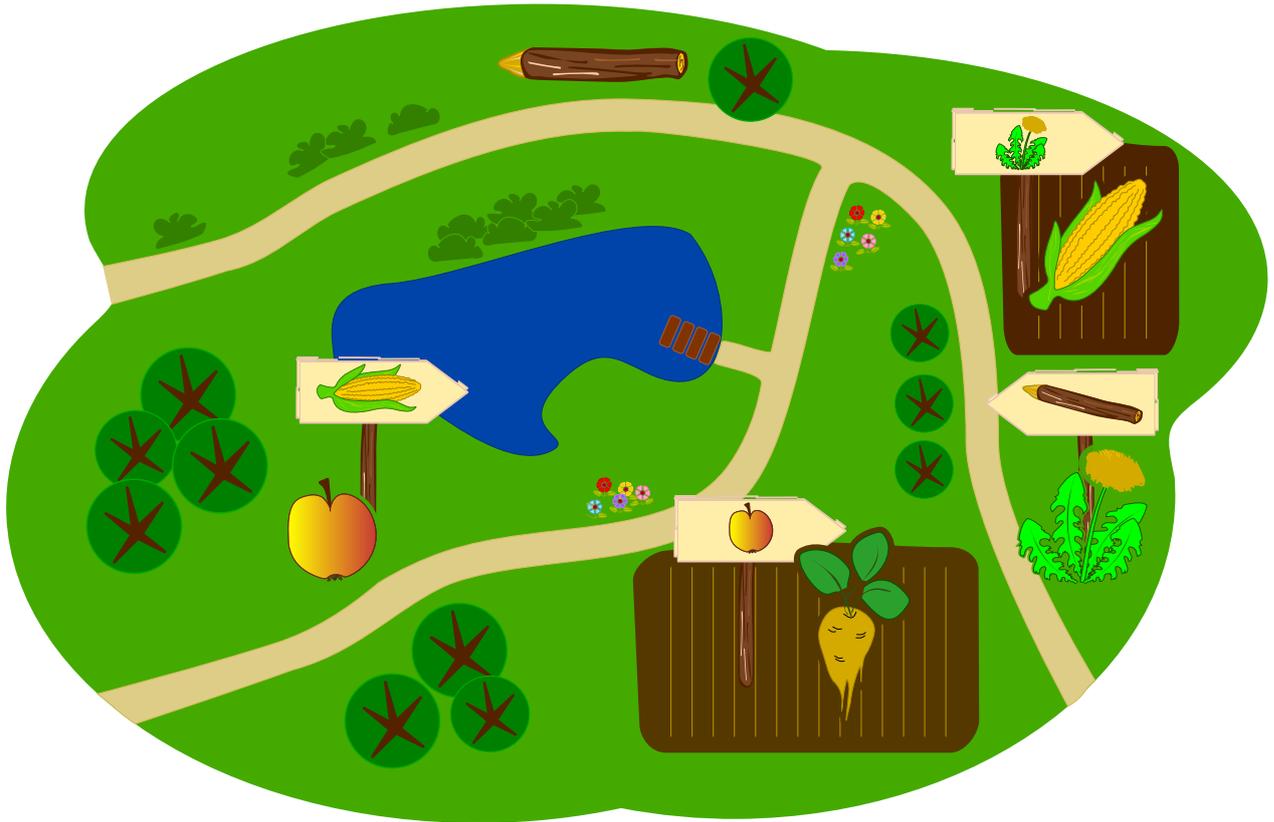




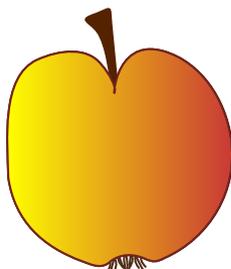
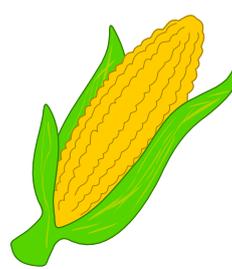
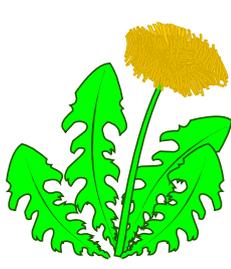
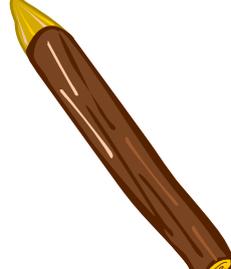
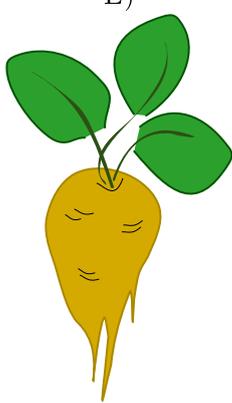
10. Une recette secrète

Les castors préparent une fête dans le jardin et ils aimeraient bien faire un gâteau type «Crunchy Cake». Malheureusement, leur chef de cuisine a pris congé. Ce qu'ils savent, c'est qu'il faut mélanger cinq ingrédients dans l'ordre correct.

Tiens! Quand ils arrivent dans le jardin, ils réalisent que le chef de cuisine les a aidés : à côté de chaque ingrédient qui pousse dans le jardin, il a planté un panneau indicateur vers l'ingrédient suivant qu'il faut mélanger avec le précédent et qui pousse également dans le jardin.



Lequel des ingrédients est-ce que les castors doivent choisir en premier ?

- A)  B)  C)  D)  E) 



Solution

La réponse correcte est E) 🍷. Le premier ingrédient ne peut être que celui qui ne se trouve pas sur un des panneaux indicateurs. Si un ingrédient se trouve sur un panneau indicateur, cela signifie qu'un autre ingrédient a dû être choisi auparavant. Si, par exemple, la pomme 🍏 était le premier ingrédient, le panneau planté dans le champ de la betterave à sucre 🍷 ne devrait pas indiquer une pomme.

C'est de l'informatique !

Le chef de cuisine des castors a déposé la recette pour le Crunchy Cake de façon particulière : pour leur indiquer l'ordre correct des ingrédients, il a relié chaque ingrédient avec l'ingrédient suivant et ceci, à l'aide de panneaux indicateurs. En informatique, le procédé qui consiste à indiquer une séquence s'appelle une *liste chaînée*. Elle est souvent utilisée quand on veut partir d'un élément initial pour aller successivement vers les autres éléments. Cependant, si on a tous les éléments dès le début et que l'on ne connaît pas l'élément initial, tout devient un peu compliqué, comme tu l'as peut-être remarqué en attaquant le problème présent. C'est la raison pour laquelle les informaticiens, en créant des listes, indiquent toujours un élément initial.

On pourrait également s'imaginer que plusieurs éléments précèdent un même élément. Si nous prenons l'exemple de notre gâteau, on préparerait ainsi en même temps, mais séparément, le fond de pâte et la crème avant de les mettre ensemble. Dans ce cas-là, l'ordre dans lequel on a préparé les deux éléments n'est pas important. Il ne s'agirait donc plus d'une liste, mais bien d'un *arbre* car on arrive à un élément final commun en parcourant de différentes *arcs* (ou *liens*).

Sites web et mots clés

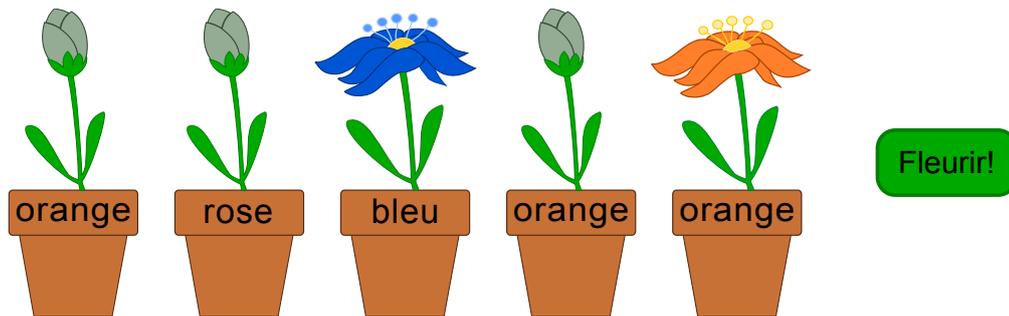
arbre, graphe, liste chaînée

— https://fr.wikipedia.org/wiki/Liste chaînée#Liste_simplement chaînée

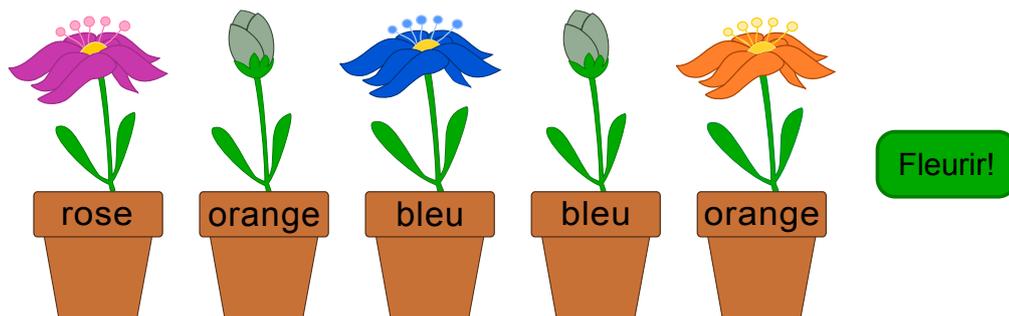


11. Que les fleurs fleurissent !

Jana joue à un jeu vidéo sur ordinateur. Secrètement, l'ordinateur a choisi les couleurs des 5 bourgeons. Les couleurs disponibles sont le bleu, l'orange et le rose. La gamme des fleurs ne peut pas être changée. Jana a choisi la couleur pour chaque bourgeon et elle a cliqué sur «Fleurir!». Seules les fleurs dont Jana a deviné la juste couleur fleuriront, les autres ne fleuriront pas.



Ensuite, Jana change les couleurs de quelques fleurs. Le résultat est le suivant :



Choisis la couleur pour chacune des fleurs.



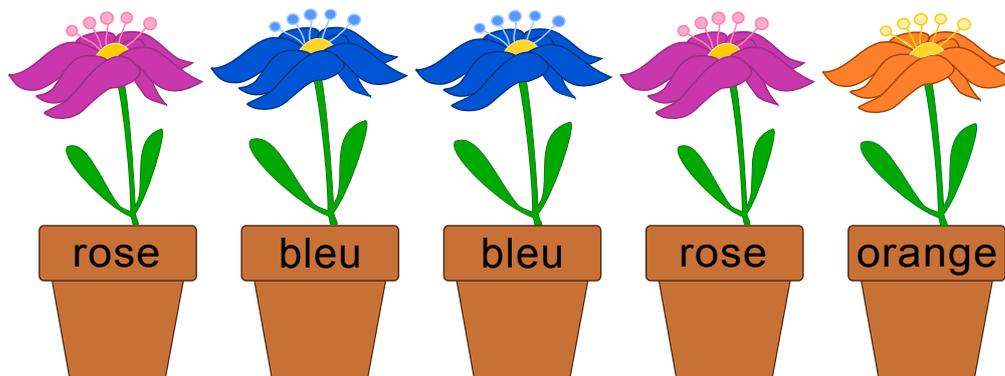
Solution

La réponse correcte est : «rose», «bleu», «bleu», «rose» et «orange».

Après avoir essayé de deviner deux fois la couleur juste, il n'y a eu que trois bourgeons à avoir fleuri. C'est la raison pour laquelle nous pouvons déjà déterminer les couleurs choisies par l'ordinateur pour le premier, le troisième et le cinquième bourgeon.

Lors du premier essai, Jana a choisi pour le deuxième bourgeons la couleur «rose» avec le résultat qu'il n'a pas fleuri. Lors du deuxième essai, elle a choisi «orange» et le bourgeon n'a toujours pas fleuri. Puisqu'il n'existe que trois couleurs, il s'ensuit que le deuxième bourgeon doit être «bleu».

Ensuite, Jana a choisi «orange» et «bleu» pour le bourgeon de la quatrième fleur et cette dernière n'a pas fleuri. Elle en déduit correctement que le quatrième bourgeon doit être «rose».



C'est de l'informatique !

Une des capacités les plus importantes dans la résolution des problèmes est de savoir tirer les bonnes conclusions des événements qui ont eu lieu ou qui, par contre, n'ont pas eu lieu. La tâche en question est une version simplifiée d'un jeu de logique très populaire. Le jeu a été simplifié dans le sens que le joueur obtient toutes les informations dont il a besoin pour trouver la solution après avoir essayé de deviner la bonne solution lui-même. Au troisième essai, au plus tard, le joueur connaîtra la couleur juste de la fleur particulière... à moins qu'il ait bien fait attention auparavant.

Sites web et mots clés

logique, jeu de stratégie, algorithme

— <https://fr.wikipedia.org/wiki/Mastermind>



12. Un anniversaire à deux valeurs

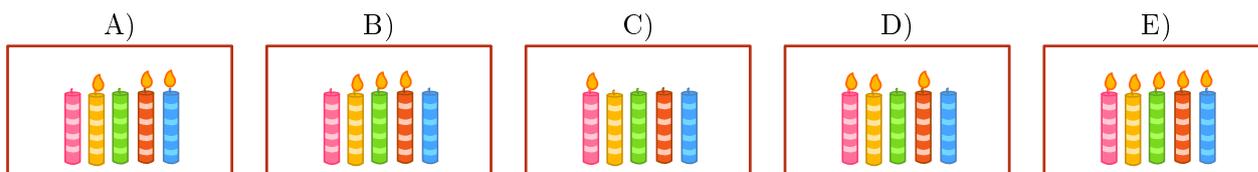
C'est l'anniversaire de Benno : il va avoir onze ans. Malheureusement, sa mère ne dispose que de cinq bougies pour le gâteau d'anniversaire. Pourtant, elle a déjà une petite idée comment représenter le nombre onze avec cinq bougies. Son plan est de planter une bougie à côté de l'autre tout en définissant les conditions suivantes :

- La bougie à l'extrême droite prend la valeur numérique 1.
- Toutes les autres bougies prennent la valeur numérique double de la bougie à leur droite.
- On n'additionne que les valeurs numériques des bougies allumées.

Voici quelques exemples :



Quelles bougies sa mère va-t-elle donc allumer pour représenter le nombre 11 ?





Solution

La réponse correcte est A) : A (01011) : la mère va allumer les bougies numéro 8, 2 et 1. Ainsi, la valeur numérique totale des bougies sera $0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$.



Voici pourquoi les réponses B), C), D) et E) sont incorrectes :

B) (01110), car si la mère allume les bougies qui correspondent aux valeurs numériques 8, 4 et 2, la valeur numérique totale sera 14 : $0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 14$.



C) (10000), car si elle n'allume que la bougie à l'extrême droite, la valeur numérique totale sera 16 : $1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 16$.



D) (11010), car si elle allume les bougies qui correspondent aux valeurs numériques 16, 8, 2, le total sera 26 : $1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 26$.



E) (11111), car si elle allume toutes les bougies, la valeur numérique totale sera 31 : $1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 31$.



C'est de l'informatique !

Grâce au système de numération binaire, on peut créer un nombre indéterminé de nombres. À l'aide de la formule «bougie éteinte» ou «bougie allumée», on définit si la valeur numérique sera additionnée ou non. L'emplacement, lui, détermine le montant de la valeur numérique. Si on remplaçait les bougies par les caractères 0 (bougies éteinte) et 1 (bougie allumée), on arriverait aux mêmes résultats. En effet, pour représenter les données, presque tous les processeurs des ordinateurs actuels utilisent le système de numération binaire (également appelé système de numération utilisant la base 2). Ceci pour des raisons pratiques : il est plus facile de réaliser des circuits logiques pour le système de numération binaire que, par exemple, pour le système décimal.



Sites web et mots clés

Système de numération binaire, représentation de l'information en binaire, système de numération utilisant la base 2.

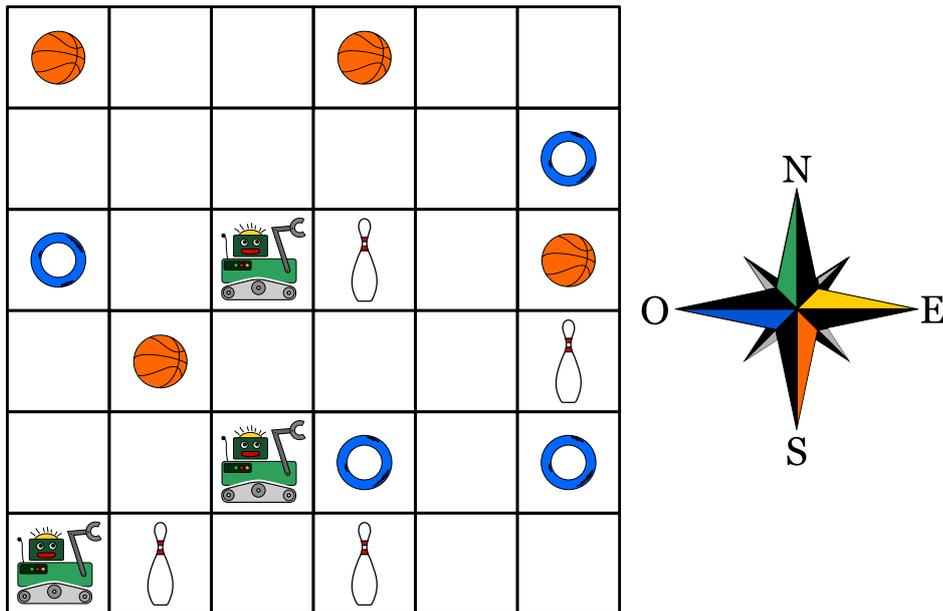
— https://fr.wikipedia.org/wiki/Système_binaire





13. Tous en même temps

Trois robots travaillent en équipe. Tu peux piloter cette équipe à l'aide des commandes de démarrage et de sens : vers le nord (N), le sud (S), l'est (E) ou l'ouest (O). En utilisant la commande de démarrage et de sens, les trois robots bougent non seulement en même temps mais aussi dans la même direction : ils avancent donc tous ensemble d'une case, et ceci dans la direction souhaitée. L'objectif est de piloter les trois robots dans la direction des objets qu'ils doivent finalement saisir. Pour qu'ils ne saisissent pas des objets non désirés, tu dois les piloter de sorte qu'ils les évitent. Exemple : Si tu pilotes les robots avec les commandes N, N, S, S, E, ils saisiront à la fin deux quilles et un anneau.



Les robots doivent saisir un ballon, un anneau et une quille.

Laquelle des quatre séries de commandes permettra aux trois robots de saisir en même temps les trois objets souhaités ?

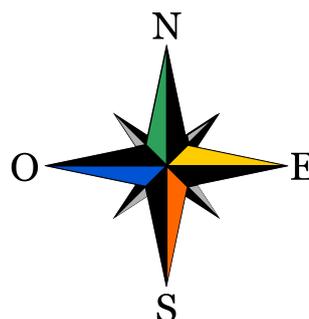
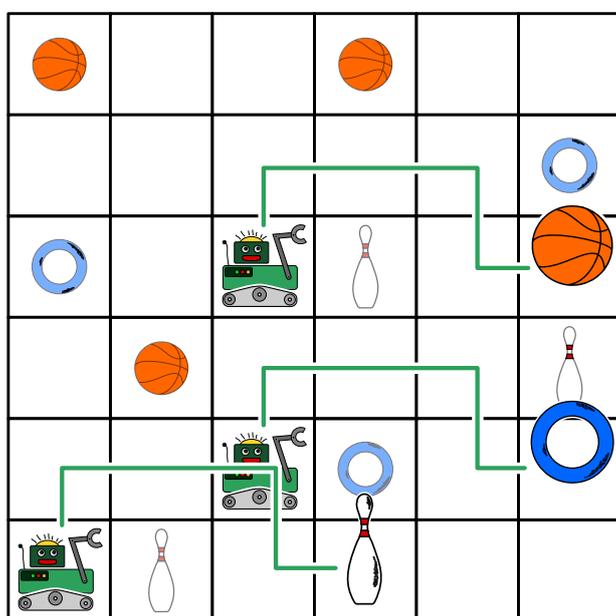
- A) N, E, E, E
- B) N, E, E, S, E
- C) N, N, S, E, N
- D) N, E, E, S, O



Solution

La réponse B) est correcte :

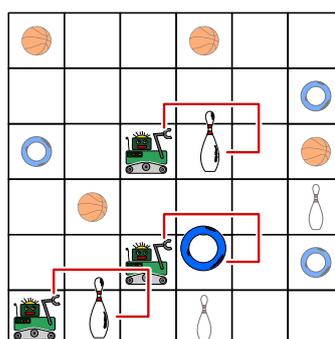
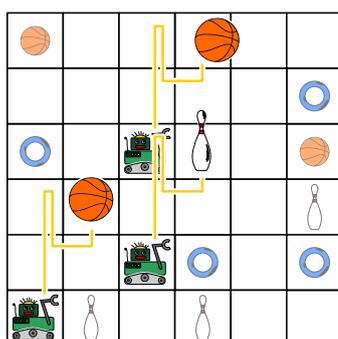
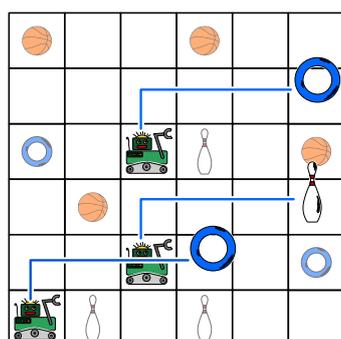
Avec les commandes N, E, E, S, E, tu pilotes les robots le long des lignes vertes. À la fin de leur parcours, les robots saisissent en effet un ballon, un anneau et une quille.



La réponse A) n'est pas correcte : avec les commandes N, E, E,E, tu pilotes les robots le long des lignes bleues. À la fin, ils saisissent deux anneaux et une quille.

La réponse C) n'est pas correcte : Avec les commandes N, N, S, E, N, tu pilotes les robots le long des lignes jaunes. À la fin, ils saisissent deux ballons et une quille.

La réponse D) n'est pas correcte : Avec les commandes N, E, E, S, O, tu pilotes les robots le long des lignes rouges. À la fin, ils saisissent deux quilles et un anneau.



C'est de l'informatique !

Dans la plupart des cas, on écrit des programmes informatiques de façon que les opérations seront exécutées successivement. De même, la plupart des langages de programmation ainsi que beaucoup de gens qui savent programmer fonctionnent de la même manière, c'est-à-dire ils exécutent une opération après l'autre.

Depuis plusieurs années déjà, le développement des processeurs touche à ses limites : il est de plus en plus difficile d'améliorer la performance des processeurs qui exécutent des programmes successivement. En même temps, on se rend compte qu'il est plus économique d'incorporer des plate-formes



de traitement multi-cœur dans un seul ordinateur. En effet, les ordinateurs actuels réunissent en règle générale 2, 4 ou plus de cœurs de processeur qui sont capables d'exécuter plusieurs opérations en même temps. Dans le cas des cartes graphiques, cette pratique est encore plus extrême : elles possèdent souvent de nombreux cœurs dont chacun d'entre eux ne satisferait jamais à la performance attendue, mais qui, dans l'ensemble, sont très performants. À part des calculs traitant la représentation des images, certains processeurs de cartes graphiques sont mêmes capables d'exécuter n'importe quelle opération demandée.

Ceci implique pourtant un changement dans l'approche de la programmation : il faut veiller à ce que les processeurs qui effectuent en parallèle différentes opérations ne se compromettent pas l'un ou l'autre, par exemple qu'ils n'attendent pas jusqu'à ce que l'autre processeur ait fini son opération, afin qu'ils terminent leur tâche ensemble tout en présentant un résultat. C'est exactement cet aspect-là qu'il faut prendre en considération dans la tâche présente.

Sites web et mots clés

programmation parallèle

- [https://fr.wikipedia.org/wiki/Parallélisme_\(informatique\)](https://fr.wikipedia.org/wiki/Parallélisme_(informatique))
- https://fr.wikipedia.org/wiki/General-purpose_processing_on_graphics_processing_units

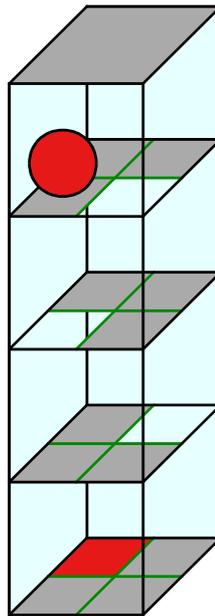
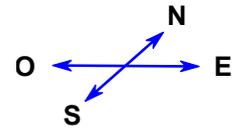




14. Le parcours de la boule

Un labyrinthe en trois dimensions se base sur quatre niveaux comptant chacun quatre champs. Une boule est placée sur le niveau le plus haut. Au niveau le plus bas se trouve la zone cible : le champ marqué en rouge.

Tu peux piloter la boule à l'aide des commandes de démarrage et de sens N (vers le nord), E (vers l'est), S (vers le sud) et O (vers l'ouest). Si la boule entre en contact avec un champ transparent, elle tombe sur le niveau inférieur. Le labyrinthe est un espace fermé ; tu ne peux donc pas piloter la boule à travers les murs.



Pilote la boule jusqu'à la zone cible !

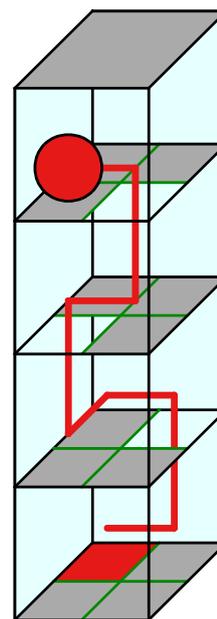


Solution

Avec la séquence de commandes comme «EONEO» (voir la ligne rouge dans la figure de droite) ou «EOENO» tu pilotes la boule jusqu'à la zone cible. Pourtant, il existe bien d'autres possibilités pour piloter la boule jusqu'à la zone cible et nul n'interdit de faire des détours ou des parcours plus longs.

C'est de l'informatique !

La séquence de commandes avec laquelle la boule est pilotée à travers le labyrinthe est en vérité un programme informatique court. Le langage y correspondant ne connaît que quatre commandes : N, O, S et E. En informatique, on parle également d'*instructions*. Un programme qui prend recours à ce langage se base sur une séquence d'instructions. Ces instructions sont exécutées de manière successive (en informatique on dit aussi de manière *séquentielle*). Les langages de programmation professionnelle prennent également recours aux séquences d'instructions. Ces dernières font partie des éléments fondamentaux pour créer un programme qui comporte de différentes instructions. Voici d'autres éléments fondamentaux pour structurer un programme informatique : la répétition (cycles), l'instruction conditionnelle (sélection) ainsi que l'appel à des parties du programme fréquemment utilisées à l'aide des sous-programmes. Et ce sera tout ; en effet, les programmes informatiques les plus complexes sont basés sur de telles structures apparemment très simples.



Sites web et mots clés

programme, séquence

— https://fr.wikipedia.org/wiki/Programmation_structurée



15. Des fleurs et des soleils

Barbara a reçu deux tampons à encre. Avec l'un des deux, elle peut créer l'empreinte d'une fleur, avec l'autre, l'empreinte d'un soleil. Maintenant, elle réfléchit comment elle pourrait créer son nom à l'aide de ces deux motifs.

Pour représenter les différentes lettres, elle choisit des suites différentes de fleurs et de soleils.

lettre	B	A	R	E	Y
suite					

À l'aide de ses suites, elle peut représenter son propre nom, «Barbara», comme suit :



Puis, elle aimerait bien représenter le nom d'un de ses amis :



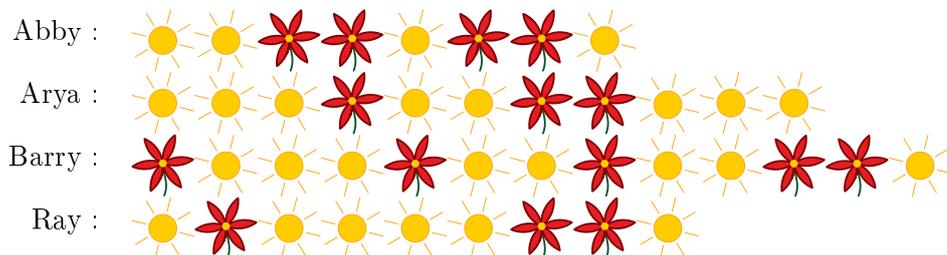
Lequel des noms ci-dessous est-ce que Barbara a représenté ?

- A) Abby
- B) Arya
- C) Barry
- D) Ray



Solution

La réponse correcte est Abby. Voici tous les codes des noms de ses amis :



C'est de l'informatique !

Il existe de différentes manières pour coder des données. Une pratique courante est par exemple d'enregistrer les caractères que l'on a tapés sur le clavier en UTF-8. L'UTF-8 est une variante de l'unicode. Pour enregistrer les caractères fréquents, on a besoin d'un espace d'un octet ce qui revient à plus de 250 caractères différents. Pour des caractères peu fréquents, il faut un espace de quatre octets ; ainsi, on peut réaliser plusieurs millions de caractères différents et ceci même dans de différentes langues.

Ce système fonctionne déjà assez bien, pourtant, il s'avère que parmi les caractères fréquents, il y en a ceux qui sont utilisés de manière plus fréquente que d'autres : le «E» ou le «N» sont par exemple plus souvent utilisés que le «X» ou le «Ö». Dans ce cas, on a recours à des codes utiles qui fonctionnent avec une longueur de symboles entièrement variable.

Dans le cas des codes à longueur variable, il est judicieux d'utiliser des débuts de codes distinctifs pour chaque symbole de la source afin de pouvoir la décoder de manière rapide et sûre. On appelle de tels codes des codes préfixe. Un des codes préfixe les plus connus est le code Morse.

Pour obtenir un code dont les données sont comprimées au maximum, il faut que l'on recherche d'abord le nombre d'occurrences de chaque caractère. Ensuite, à l'aide du codage dit codage de Huffman, on arrive à calculer un code dont les données sont comprimées au maximum. Chaque code de Huffman est également un code préfixe.

Sites web et mots clés

code préfixe, code de Huffman, compression de données

- https://fr.wikipedia.org/wiki/Code_préfixe
- https://fr.wikipedia.org/wiki/Codage_de_Huffman



16. Le tchat des castors

Le tchat nommé «Tchat des castors» peut être utilisé gratuitement et sa diffusion est financée par la publicité. L'agence de voyage «Sunshine Travel» profite de cette plate-forme et elle y est présente avec de différentes images publicitaires afin d'atteindre de différents groupes cibles. Pour choisir les images, elle profite des résultats de l'analyse des messages du tchat. Au centre de ses intérêts se trouvent des mots particuliers auxquels le système attribue automatiquement des points positifs ou négatifs :

- Les formules de salutation comme «Chère(s)» et «Cher(s)» sont particulièrement utilisés par des «castors» âgés. Le système les pénalise de deux points négatifs.
- Les formules de salutation comme «Hi», «Hey» ou «Arriba» sont très populaires parmi les jeunes castors. Le système attribue à ces messages-là deux points positifs.
- Aux messages comprenant les abréviations telles que «bf» (meilleur(e)s ami(e)s), «gr8» (merveilleux, fantastique), «np» (aucun problème) ou «thx» (merci), le système attribue un point positif.
- Chaque message avec des mots composés de 10 lettres ou plus est sanctionné par un point négatif.

Ainsi, chaque castor sera assigné à un groupe cible selon le total des points qui ont été attribués à ses messages.

nombre de points	groupe cible	image affichée
inférieur à 0	des personnes âgées	
supérieur à 0	des jeunes	
0	aucun des deux groupes cible	

Lisez les messages suivants. Quelles images l'agence de voyages choisira-t-elle pour chacun des messages ?

- A) Chers amis, l'été approche et je cherche un logement sympa tout près du Rhin. Merci de vos suggestions, Richie.
- B) Arriba! Y a-t-il quelqu'un ?
- C) @Mia : <3 <3 <3
- D) d'acc. gr8. Thx



Solution

Message A) : Sunshine Travel affiche l'image de la plage. Le message comporte la formule de salutation «Chers» et le mot «suggestions» qui, lui, est composé de plus de 10 lettres. Le système attribue donc à ce message un point négatif.

Message B) : Sunshine Travel affiche l'image avec les planches de surf. Le message comporte la formule de salutation très populaire auprès des jeunes castors.

Message C) : Sunshine Travel affiche l'image de la tour Eiffel, le synonyme pour des voyages à la découverte d'une ville. Le système attribue à ce message zéro points car aucune des règles préalablement établies ne peut y être appliquée.

Message D) : Sunshine Travel affiche l'image avec les planches de surf. Le message comporte les deux abréviations «gr8» et «thx» et le système lui attribue un point positif.

C'est de l'informatique !

La valorisation des textes selon des règles préalablement établies peut être facilement exécutée à l'aide de programmes informatiques. La recherche d'éléments particuliers dans un texte est un exemple simple de recherche basée sur le modèle appelé «pattern matching» ou filtrage par motif. Ce procédé de filtrage trouve son application aussi bien dans le traitement de texte que dans d'autres domaines comme par exemple dans le traitement des images ou encore dans d'autres programmes informatiques.

De nombreuses entreprises ont recours à l'évaluation automatisée des profils des utilisateurs sur Internet afin de créer des offres adaptées au type de client et à ses besoins. En tant qu'internaute, il s'avère donc judicieux de prendre conscience des procédés décrits ci-dessus et de gérer ses données personnelles de manière responsable. Du coup, les informaticiens se retrouvent face à un dilemme : d'une part, ils doivent aider à aiguïser la conscience du public envers de telles démarches, d'autre part, ce sont justement ces démarches qui créent de nouveaux emplois dans le secteur de l'informatique. Pour les informaticiens, les fonctions de valorisation des textes peuvent être également très utiles dans d'autres domaines, comme par exemple pour afficher les résultats de recherche selon leur pertinence.

Sites web et mots clés

profilage des utilisateurs, publicité appropriée aux groupes cible

— https://fr.wikipedia.org/wiki/Publicité_en_ligne



17. Quatre courses à faire

Alexandra aimerait bien faire les courses pendant la pause de midi (de 12h 00 jusqu'à 13h 00) :

- aller acheter un livre à la librairie ;
- aller acheter une bouteille de lait à l'épicerie ;
- envoyer le livre qu'elle vient d'acheter à la librairie par poste ;
- aller boire un café au restaurant.

Alexandra a calculé le temps dont elle a besoin pour effectuer chaque achat. Les durées listées ci-dessous ne comptent que pour des situations en dehors des heures de pointe. C'est la raison pour laquelle Alexandra essaie de les éviter à tout prix.

	Lieu	Durée	Heures de pointe
	Librairie	15 minutes	12h 40 – 13h 00
	Épicerie	10 minutes	12h 00 – 12h 40
	Poste	15 minutes	12h 00 – 12h 30
	Restaurant	20 minutes	12h 30 – 12h 50

Déplace les achats divers en une séquence correcte afin qu'Alexandra puisse éviter les heures de pointe.



Solution

La séquence correcte est : restaurant, librairie, poste, épicerie.

Ce problème a quelques limitations que l'on peut rendre évident à l'aide du tableau suivant : (en rouge foncé : les heures de pointe, en vert clair : les heures normales) :

Lieu	Durée	12.00-12.05	12.05-12.10	12.10-12.15	12.15-12.20	12.20-12.25	12.25-12.30	12.30-12.35	12.35-12.40	12.40-12.45	12.45-12.50	12.50-12.55	12.55-13.00
Librairie	15 minutes					X	X	X					
Épicerie	10 minutes											X	X
Poste	15 minutes								X	X	X		
Restaurant	20 minutes	X	X	X	X								

Alexandra doit avoir quitté la librairie avant 12h 40 et elle ne peut aller à l'épicerie qu'après 12h 40. Aller à la poste n'est possible qu'après avoir été à la librairie. Mais elle ne peut aller à la poste qu'après 12h 30 et elle doit avoir bu un café avant 12h 30 parce que sinon, à 12h 50, elle n'aura pas assez de temps pour sa pause de midi.

Voilà le seul plan possible pour qu'elle puisse faire tous les achats tout en évitant les heures de pointe (dans le tableau marqué par X) :

- aller boire un café au restaurant entre 12h 00 et 12h 20 ;
- aller acheter un livre à la librairie entre 12h 20 et 12h 35 ;
- aller à la poste pour envoyer le livre qu'elle vient d'acheter à la librairie entre 12h 35 et 12h 50 ;
- aller à l'épicerie entre 12h 50 et 13h 00.

C'est de l'informatique !

Quand on doit résoudre un problème, l'informatique poursuit un objectif important qui est de trouver des solutions qui répondent le mieux aux limitations (conditions) existantes. Une des conditions de notre problème est qu'Alexandra doit éviter les heures de pointe. Souvent, dans d'autres situations où il faut résoudre un problème, la question est en effet de savoir s'il existe réellement une solution, plus précisément si on peut trouver une solution qui réponde à toutes les conditions préalablement définies.

En informatique, ce type de questionnement s'appelle «scheduling» (planification). «Scheduling» signifie «planification de tâches informatiques», c'est-à-dire qu'on cherche à trouver la séquence la plus correcte et la plus favorable pour exécuter des opérations d'une tâche particulière. Le scheduling trouve son application, par exemple, dans le domaine de l'industrie, lors des projets vastes ou bien dans la production des pièces industrielles. On a également recours au scheduling pour les ordinateurs, par exemple, quand différents processeurs (ou CPU) dotés de plusieurs cœurs doivent exécuter différentes opérations en même temps.

Sites web et mots clés

planification de tâches informatiques, optimisation

- https://fr.wikipedia.org/wiki/Ordonnancement_de_tâches_informatiques



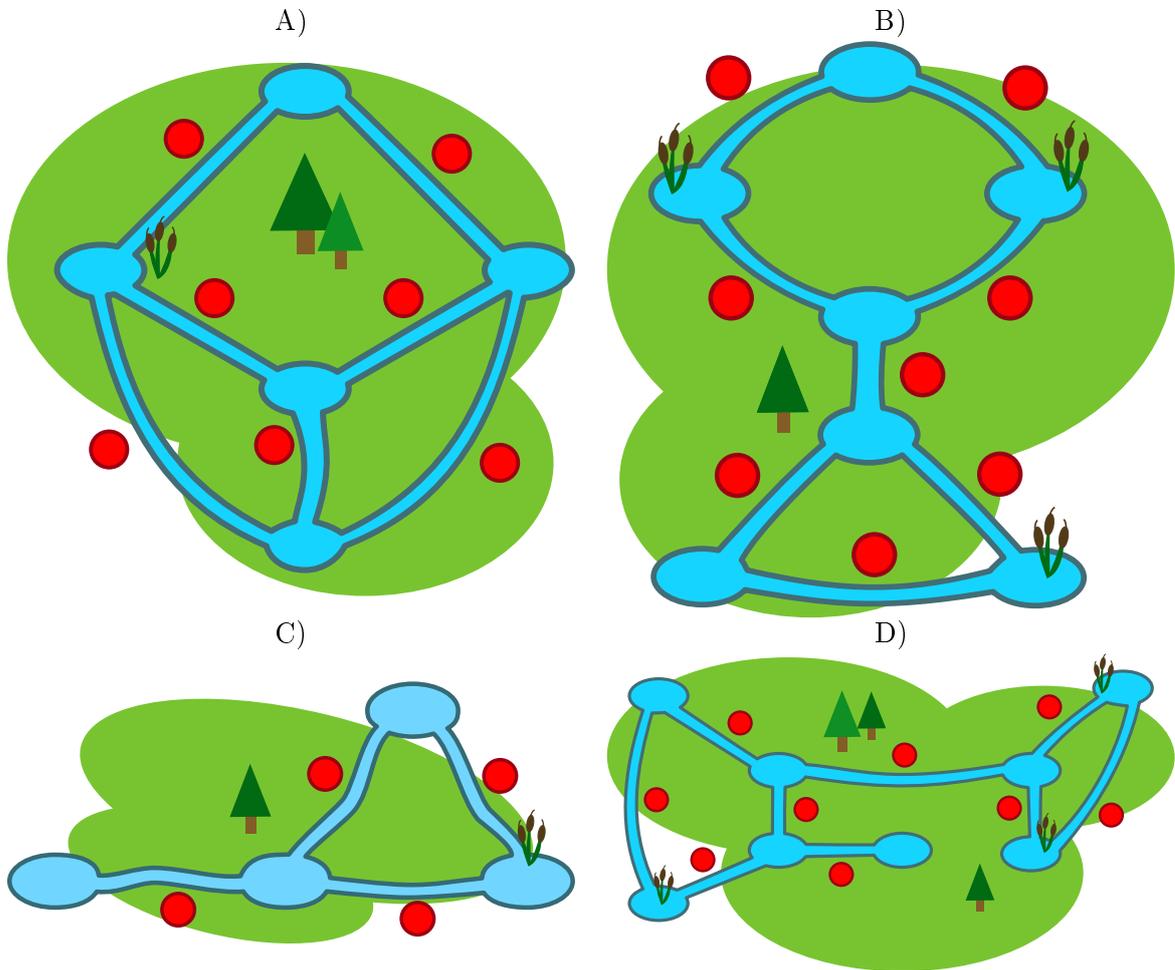
18. Des impasses

Les castors adorent construire des barrages. Mais quand ils nagent dans leurs canaux et qu'ils arrivent à un de ces barrages, ils sont obligés de faire un détour en sortant de l'eau. Voilà quelque chose qu'ils n'aiment pas du tout ! Au lieu de sortir de l'eau, ils préfèrent rebrousser chemin et choisir un autre canal afin d'atteindre leur destination finale.

Les castors appellent «impasses» les canaux qu'ils doivent absolument parcourir parce qu'ils n'ont pas d'autre choix pour arriver à leur destination finale. Ils décident donc de ne pas construire de barrages dans les impasses.

Sur les images ci-dessous, on aperçoit quelques territoires de castors. Les repères rouges marquent l'endroit approximatif des lieux de construction des barrages dans le canal.

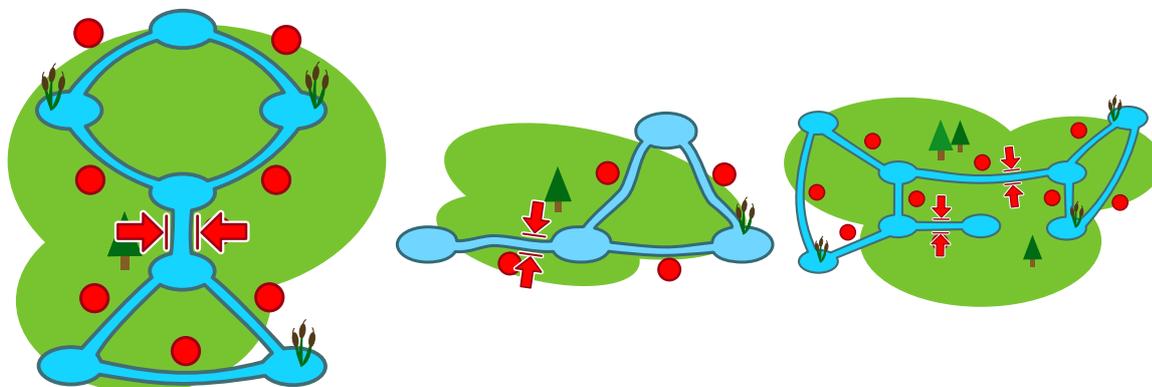
Il n'y a qu'un seul territoire qui n'a pas d'impasse. Il s'agit duquel ?





Solution

La réponse correcte est A). Dans ce territoire, il existe pour chaque canal un chemin alternatif. Dans le territoire B), l'impasse se trouve au centre. Dans le territoire C), l'impasse se trouve à l'extrême gauche et dans le territoire D) il y a deux impasses qui se trouvent au centre.



C'est de l'informatique !

Le territoire dans lequel vivent les castors est un réseau comprenant des canaux et des étangs. C'est tout à fait comparable à l'Internet : ainsi, les étangs représenteraient les ordinateurs, les portables, les téléviseurs etc. et les canaux représenteraient les lignes de transmissions ou les connexions sans fil. À l'origine, l'Internet a été inventé pour connecter les universités des États-Unis entre elles. Ses fondateurs déjà, veillaient à éviter les impasses car ils savaient qu'au cas où une connexion poserait des problèmes, il n'existerait pas d'alternative.

Pour observer des problèmes de réseau, l'informatique se sert de la théorie des graphes. Les graphes sont définis comme des systèmes de nœuds (étangs) et des arcs (canaux). À l'aide des graphes, il est possible de modéliser différents types de réseaux comme par exemple un réseau de transport ou de communication. Pour analyser des problèmes de réseau, on a développé beaucoup d'algorithmes. Un problème bien connu est celui de trouver des « ponts » à l'intérieur des graphes, c'est-à-dire de trouver les impasses indésirables à l'intérieur du réseau – tout comme il fallait le faire dans la tâche présente.

Sites web et mots clés

graphes, ponts, impasses

— [https://fr.wikipedia.org/wiki/Isthme_\(théorie_des_graphes\)](https://fr.wikipedia.org/wiki/Isthme_(théorie_des_graphes))



19. Des messages secrets

Les agents Boris et Bertha communiquent à travers des messages secrets que personne n'est censé comprendre. Boris aimerait bien envoyer à Bertha le message secret suivant :

RENCONTRERBILLYA6H

Il écrit les lettres du texte successivement dans un tableau à quatre colonnes et cinq rangées. Il commence à gauche en haut et continue, case après case, rangée après rangée jusqu'à ce qu'il arrive en bas. Si jamais il reste à la fin quelques champs libres, il insère un astérisque par champ libre. Voici le résultat :

R	E	N	C
O	N	T	R
E	R	B	I
L	L	Y	A
6	H	*	*

Ensuite, il crée le message secret. Sur une feuille vierge, il réécrit les lettres du tableau ci-dessus de haut en bas, case après case, colonne après colonne, de gauche à droite.

ROEL6ENRLHNTBY*CRIA*

Bertha reprend cette méthode pour créer sa réponse. Elle lui envoie le message secret suivant :

OVD AIKIRU*JEAS*ENIS*

Choisis le message secret que Bertha a envoyé à Boris.

- A) OKJEVERR AISIJEPEUX
- B) OKJEVIENDRAIAUSSI
- C) OKJESERAILAAUSSI
- D) OKJENEPEUXPASVENIR



Solution

Voilà comment on peut reconstruire le texte original : on insère à nouveau le message secret reçu dans un tableau à quatre colonnes et cinq rangées, en commençant à gauche, tout en haut, pourtant cette fois-ci, on continue de haut en bas, colonne par colonne.

O	K	J	E
V	I	E	N
D	R	A	I
A	U	S	S
I	*	*	*

Quand on lit le texte de gauche à droite, rangée par rangée on obtient le message suivant :

OKJEVIENDRAIAUSSI

Les astérisques à la fin du message ne font plus partie du message.

C'est de l'informatique !

Les messages que nous transmettons à travers un réseau informatique peuvent être facilement interceptés. Quand ces messages contiennent des données sensibles comme des mots de passe ou des informations privées, nous préférons qu'il n'y ait que le destinataire qui puisse lire le texte. Dans ce cas-là, on peut coder le message (le texte en langage clair) afin qu'il soit transformé en un message secret. Seul le destinataire saura comment déchiffrer ce message pour obtenir le message original.

Dans la pratique, il existe de différentes méthodes de cryptage. La méthode que l'on vient d'utiliser dans cette tâche est appelée transposition et on l'applique depuis environ 2'400 ans. En appliquant cette méthode, on maintient toutes les lettres du texte clair. On ne change que leur ordre. Cette méthode est malheureusement facile à déchiffrer, en fait, il ne s'agit même pas d'une véritable méthode de cryptage. On parlerait plutôt d'un camouflage de l'information.

La cryptographie est une technique d'écriture en langage chiffré ou codé. En tant que science, elle étudie les méthodes et technologies de cryptage et représente un domaine important de l'informatique. Toutes les entreprises commerciales ainsi que les banques qui sont présentes sur Internet dépendent des systèmes de chiffrement hautement sécurisés. Les chiffrements modernes sont exécutés par ordinateur et se basent sur des méthodes mathématiques qui rendent pratiquement impossible de déchiffrer un message si on ne connaît pas la clé.

Sites web et mots clés

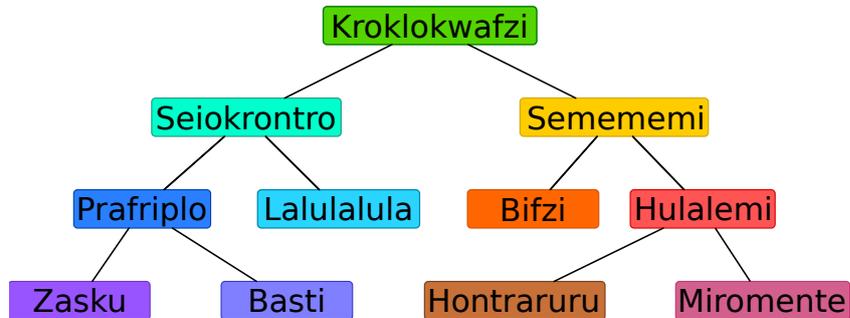
chiffrement, cryptographie, transposition, protection des données

— https://fr.wikipedia.org/wiki/Chiffrement_par_transposition



20. Vive la hiérarchie !

L'image ci-dessous représente les relations entre les espèces vivant sur la planète Morgenstern. On interprète la ligne qui relie deux espèces de telle manière que tous les animaux de l'espèce inférieure font également partie de l'espèce notée au-dessus.



Exemple : tous les «Hulalemi» font partie de l'espèce des «Semememi». Certains «Seiokronro», par contre, ne font pas partie de l'espèce «Basti».

Des affirmations ci-dessous, il n'y en a qu'une qui est juste. Laquelle ?

- A) Tous les Basti font partie de l'espèce Seiokronro.
- B) Certains Hontraruru ne font pas partie des Semememi.
- C) Tous les Zasku font également partie des Bifzi.
- D) Tous les Prafriplo font également partie des Basti.



Solution

La réponse A) est correcte.

A) : Tous les animaux de l'espèce Basti sont également des animaux de l'espèce Prafriple. Comme tous les animaux des Prafriple font partie de l'espèce Seiokrontro, tous les animaux des Basti font également partie de l'espèce Seiokrontro.

B) : Les Hontraruru sont des animaux de l'espèce Hulalemi. Ces-derniers sont des animaux de l'espèce Semememi. C'est la raison pour laquelle tous les animaux de l'espèce Hontraruru font partie de l'espèce Semememi (et non de l'espèce Seiokrontro).

C) : Les Zasku sont des animaux de l'espèce Prafriple. Ils ne font pas partie de l'espèce Bifzi.

D) : Les animaux de l'espèce Basti font partie de l'espèce Prafriple, et non l'inverse.

C'est de l'informatique !

Les relations entre les espèces sont représentées selon un principe que l'on appelle en informatique «arbre binaire». Les biologistes, eux, utilisent l'arbre «phylogénétique» pour illustrer les relations entre les espèces diverses.

En informatique, on a souvent recours à de tels arbres pour des représentations graphiques des relations. Un arbre généalogique représente, par exemple, la relation entre les enfants, les parents et les grands-parents. Si on utilise des arbres pour la représentation des relations, on peut rapidement comprendre quelles relations existent entre les différents éléments de l'arbre appelés des «nœuds».

Les arbres sont en outre un excellent moyen pour enregistrer des données de manière ordonnée et donc pour les retrouver le plus rapidement possible... ainsi, il ne faudra que peu d'étapes pour pouvoir accéder à un énorme volume de données.

Sites web et mots clés

arbres, spécialisation, généralisation

— [https://fr.wikipedia.org/wiki/Arbre_\(graphe\)](https://fr.wikipedia.org/wiki/Arbre_(graphe))

— https://fr.wikipedia.org/wiki/Arbre_phylogénétique

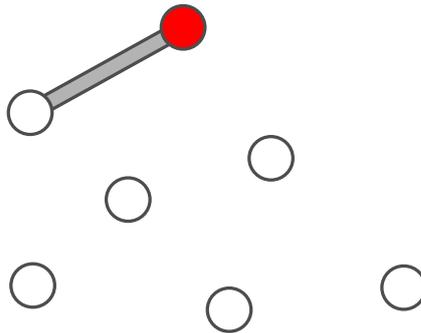
21. Plongeons-nous dans la construction des ponts!

Le grand-papa des castors est devenu un peu hydrophobe. Afin de pouvoir éviter l'eau le plus souvent possible, il imagine de relier sa hutte de castor avec toutes les autres huttes de sa famille par des ponts. Les autres castors apprécient son idée et ils décident de l'aider à condition que l'on tienne compte des points suivants :

- Pour atteindre chacune des autres huttes, grand-papa ne doit traverser que deux ponts au maximum.
- À part le pont que l'on construit pour atteindre une hutte, on ne construira que deux autres ponts au maximum pour atteindre une des autres huttes.

Avant de commencer la construction, les castors dessinent un plan. Ils décident de représenter les huttes sur le plan par des cercles. La hutte de grand-papa est représentée par un cercle rempli de couleur rouge. Quand les castors commencent, ils arrivent à dessiner un premier pont qui prend son départ de la hutte de grand-papa. Malheureusement, à partir de ce moment-là, ils ne savent plus comment continuer.

Complète le plan de sorte que toutes les conditions requises seront réunies. Il existe plusieurs possibilités pour accomplir cette tâche. Ce qui est clair, c'est que l'on a besoin de cinq autres ponts pour relier toutes les huttes.

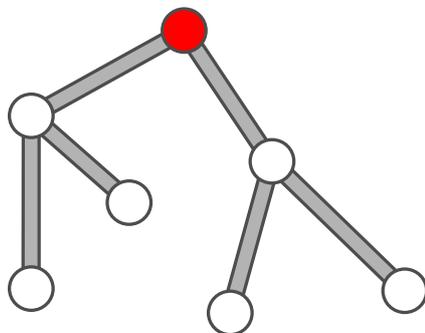




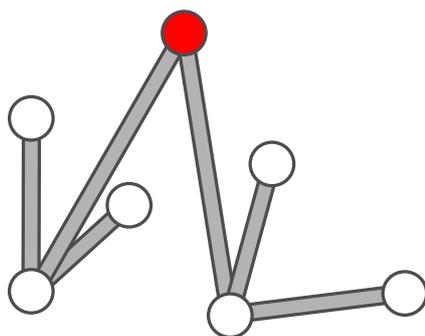
Solution

Le plan ci-dessous réunit toutes les conditions requises car :

- partant du cercle rouge, tous les autres cercles peuvent être atteints et ceci, en parcourant au maximum deux liens ;
- de chaque cercle il n'y a que trois liens qui sortent.



Bien entendu, il existe encore d'autres plans qui répondent aux conditions requises, comme par exemple le plan ci-dessous :



C'est de l'informatique !

Bien qu'il existe beaucoup de solutions possibles, chacune d'entre elles présente la même structure : de la hutte du grand-père sortent deux ponts qui la relient avec deux autres huttes. De ces dernières, deux autres ponts par hutte sortent pour en relier les quatre huttes restantes. S'il y avait une hutte de plus, il ne serait pas possible de l'atteindre tout en répondant aux conditions requises.

Avec leur plan, les castors créent une structure de données sous la forme d'une hiérarchie appelée *arbre* : tous les *nœuds* (les cercles/les huttes) peuvent être atteints en parcourant les *arcs* (les liens/les ponts). Les deux conditions préalablement définies garantissent que les castors créent un arbre particulier : étant donné qu'il n'y a que deux arcs qui doivent sortir d'un seul nœud, les castors génèrent avec leur plan un *arbre binaire*. Le fait que l'on ne doit parcourir que deux arcs au maximum pour arriver aux prochains nœuds garantit que l'arbre est *minimal*.

Sites web et mots clés

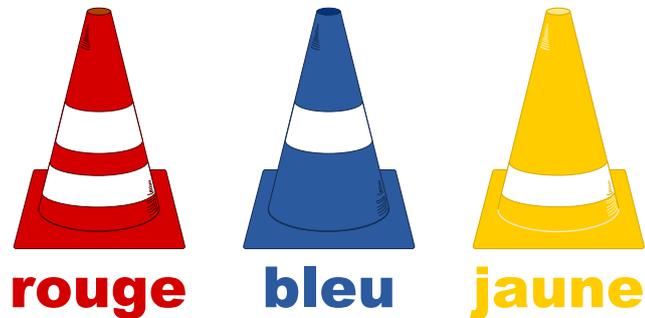
arbre, huttes

- [https://fr.wikipedia.org/wiki/Arbre_\(graphe\)](https://fr.wikipedia.org/wiki/Arbre_(graphe))
- https://fr.wikipedia.org/wiki/Arbre_binaire



22. Des cônes servant de cachette

Vreni cache des cartes à jouer sous trois cônes de circulation. Elle dépose toujours une carte sous un cône particulier. S'il y a déjà une carte, elle met la nouvelle carte au-dessus de la précédente.



Pour mémoriser laquelle des cartes à jouer se trouve sous quel cône de circulation, elle note comme suit :

rouge + 5

rouge + 3



Vreni a noté :

rouge + 3

jaune + 5

rouge + 6

jaune + 8

bleu + 1

jaune + 3

Selon les notes de Vreni, quelle carte à jouer se trouvera tout en haut de la pile sous le cône de circulation correspondant ?

- A) rouge : 3, bleu : 1, jaune : 5
- B) rouge : 9, bleu : 1, jaune : 16
- C) rouge : 6, bleu : 1, jaune : 3
- D) rouge : 8, bleu : 1, jaune : 3



Solution

Au début, il n'y a aucun cône de circulation qui cache une carte. Le tableau ci-dessous montre quelles cartes se trouvent sous quel cône tout en haut de la pile après chaque étape.

Note	rouge	bleu	jaune
	-	-	-
rouge ← 5	5	-	-
jaune ← 5	5	-	5
rouge ← 6	6	-	5
jaune ← 8	6	-	8
bleu ← 1	6	1	8
jaune ← 3	6	1	3

Ainsi, il est évident que la réponse C) est juste.

C'est de l'informatique !

Un cône peut être interprété comme une variable dont le nom est «rouge», «bleu» ou «jaune». En informatique, une variable est un espace de stockage où l'ordinateur peut sauvegarder des valeurs comme dans notre exemple des nombres entiers. Chaque commande est en effet une affectation (informatique). Cela signifie que la valeur de la variable utilisée auparavant sera substituée par une nouvelle valeur.

Dans un langage de programmation, on note les affectations comme suit : «=» ou «:=». Souvent, ces annotations peuvent être sources de raisonnements erronés : pendant qu'en mathématiques l'équation « $x = x + 1$ » est simplement fausse, en informatique, cette annotation signifie que la valeur de la variable x est augmentée de la valeur 1 et que le résultat sera sauvegardé en tant qu'une nouvelle valeur de la variable x égal à « $x + 1$ ». En dehors des langages de programmation et pour mieux différencier les affectations, on a recours à la flèche «←». Ainsi, l'annotation en question sera immédiatement comprise et par les mathématiciens et par les informaticiens.

Sites web et mots clés

variable, affectation (informatique)

- [https://fr.wikipedia.org/wiki/Variable_\(informatique\)](https://fr.wikipedia.org/wiki/Variable_(informatique))
- [https://fr.wikipedia.org/wiki/Affectation_\(informatique\)](https://fr.wikipedia.org/wiki/Affectation_(informatique))



23. Le travail en groupe

Sarah et ses camarades de classe, Alicia, Beat, Caroline, David et Emil doivent faire un travail en groupe. Ils répartissent les tâches et le groupe confie à Sarah la tâche de rassembler les résultats de tous les membres du groupe. Elle n'a aucun problème à obtenir les résultats d'Emil car il peut les lui donner immédiatement. Pour obtenir les résultats des autres camarades, par contre, elle doit respecter les conditions suivantes :

- Pour obtenir les résultats de David, Sarah doit lui présenter d'abord les résultats d'Alicia.
- Pour obtenir les résultats de Beat, elle doit lui présenter d'abord les résultats d'Emil.
- Pour obtenir les résultats de Caroline, elle doit lui présenter d'abord les résultats de Beat et de David.
- Pour obtenir les résultats d'Alicia, elle doit lui présenter d'abord les résultats de Beat et d'Emil.

Déplace les noms afin de les classer dans un ordre logique qui permettra à Sarah d'obtenir les résultats de tous les membres du groupe.





Solution

La réponse correcte est :

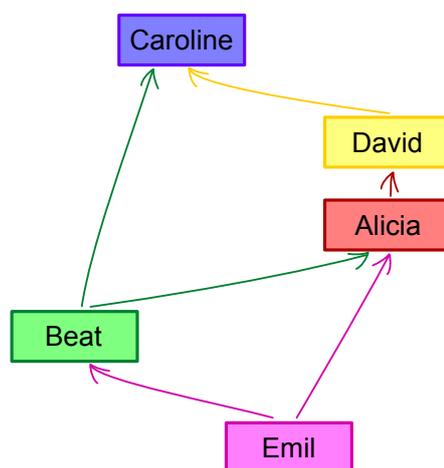


Emil est le seul qui ne dépend pas des résultats des autres. C'est la raison pour laquelle Sarah doit l'interroger en premier. Beat est le seul qui a besoin des résultats d'Emil afin de pouvoir terminer son travail. Pour qu'Alicia puisse rendre ses résultats, elle doit tenir compte des résultats d'Emil et de Beat. David ne peut terminer son travail que s'il jette un œil sur les résultats d'Emil, de Beat et d'Alicia (bien qu'en vérité, il ne s'intéresse qu'aux résultats d'Alicia). Finalement, il y a Caroline qui montrera volontiers ses résultats après avoir vu les résultats de Beat et de David.

C'est de l'informatique !

Dans la vie quotidienne, il arrive souvent que l'on ne puisse terminer un travail avant que certaines conditions préalables soient remplies. L'informatique s'intéresse également aux opérations de planification des tâches («scheduling»). La mise en œuvre de tels programmes est très fréquente dans le domaine de l'industrie. Ainsi, il est possible de réaliser des opérations complexes comme par exemple le montage d'un véhicule le plus efficacement possible (sans, par exemple, devoir attendre trop longtemps pour pouvoir monter les pièces nécessaires). En optimisant les étapes différentes au sein d'une entreprise, on économise beaucoup d'argent car on n'aura par exemple plus besoin d'un entrepôt et les machines seront plus performantes.

Pour modéliser une planification, on utilise des graphes dont les nœuds représentent les différentes opérations (dans le cas présent, l'action de demander les résultats auprès des camarades) ainsi que les arcs, ici, sous forme de flèches. Le fait d'avoir recours à des flèches signifie que les arcs sont orientés. Il s'ensuit qu'un tel graphe est appelé « graphe orienté ». Le graphe qui correspond à notre problème est représenté comme suit :



Une solution possible d'un plan de déroulement est un parcours à travers le graphe qui passe par tous les nœuds. Dans l'exemple de notre graphe, il n'y a qu'un seul parcours valable à savoir la réponse correcte à la question qui est : Emil → Beat → Alicia → David → Caroline.

Comme expliqué plus haut, la recherche d'une solution n'est qu'une partie de la solution du problème. Il existe bien d'autres situations complexes dans lesquelles les opérations se déroulent parallèlement. Il est en outre souvent important de trouver une solution qui permette de maintenir les coûts aussi bas que possible ou qui rendent l'exécution d'une opération particulièrement rapide. Dans de maintes



situations, il est préférable d'appliquer des algorithmes heuristiques (algorithme d'approximation) pour pouvoir s'approcher d'une solution optimale au lieu de simplement tester toutes les solutions possibles.

Sites web et mots clés

planification des tâches, plan de déroulement, graphe orienté

— https://fr.wikipedia.org/wiki/Ordonnancement_de_travaux_informatiques



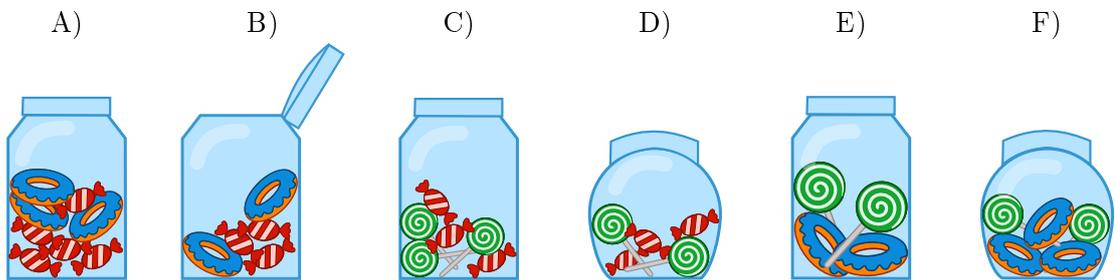


24. Des bonbonnières

Une bonbonnière est un récipient en verre utilisé pour conserver des friandises. Carl et Judy possèdent trois bonbonnières. Les bonbonnières A), B) et C) appartiennent à Carl tandis que les bonbonnières D), E) et F) sont à Judy. Chaque bonbonnière présente les cinq caractéristiques suivantes :

- La bonbonnière est soit ouverte soit fermée.
- La bonbonnière contient des bonbons à rayures blancs et rouges ou pas.
- La bonbonnière contient des bonbons-cercles bleus au sucre ou pas.
- La bonbonnière contient des sucettes en spirales et de couleur verte ou pas.
- La bonbonnière est soit ronde soit carrée.

Choisis la bonbonnière qui répond aussi bien aux caractéristiques communes des bonbonnières de Carl qu'aux caractéristiques communes des bonbonnières de Judy.





Solution

Les bonbons-cercles bleus au sucre ne doivent pas être pris en considération parce qu'ils ne constituent pas une caractéristique commune des bonbonnières de Carl et de Judy : ils n'existent simplement pas dans les bonbonnières C) et D).

Les deux caractéristiques communes des bonbonnières A), B) et C) de Carl sont les suivantes :

- chaque bonbonnière est carrée ;
- chaque bonbonnière contient des bonbons à rayures blanches et rouges.

Les deux caractéristiques communes des bonbonnières D), E) et F) de Judy sont les suivantes :

- chaque bonbonnière est fermée ;
- chaque bonbonnière contient des sucettes en spirales et de couleur verte.

Seule la bonbonnière C) réunit toutes les quatre caractéristiques : elle est carrée, elle contient des bonbons à rayures blanches et rouges ainsi que des sucettes en spirales et de couleur verte et elle est fermée.

C'est de l'informatique !

Dans leurs modèles de données, les informaticiens préfèrent regrouper les objets selon leurs caractéristiques (le terme le plus approprié est : «propriété» ou «attribut»). La présente tâche comprend cinq propriétés et deux groupes. L'objectif est de chercher des objets qui réunissent toutes les propriétés communes aux deux groupes. En relation avec des bases de données relationnelles, une telle opération est aussi appelée : «calculer l'intersection de deux ensembles».

Dans beaucoup de bases de données relationnelles, on peut identifier («filtrer») des objets à l'aide de propriétés déterminées et ceci, même parmi des quantités énormes de données («construire un sous-ensemble»). De cette manière, il est possible d'entamer une recherche ciblée sur des boutiques en ligne, par exemple. Les propriétés d'un smartphone telles que la durée de vie de la batterie, la taille d'écran et d'autres encore sont enregistrées dans une base de données à laquelle le moteur de recherche peut accéder pour permettre une recherche rapide.

Lors de la création d'une base de données, il est très important d'être particulièrement attentif aux propriétés utilisées dans le modèle de données. Si, par exemple, on oublie quelques propriétés importantes, toute recherche ultérieure sera moins précise. Si, par contre, on introduit des propriétés redondantes, l'entrée des données ultérieures sera coûteuse sans que la valeur de la base de données augmente.

Sites web et mots clés

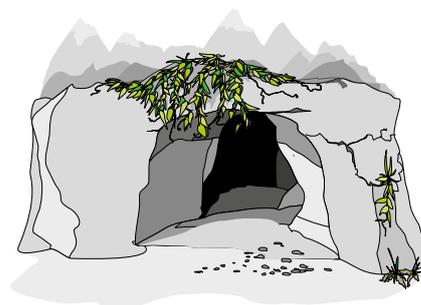
orientation objet, attribut, logique

- [https://fr.wikipedia.org/wiki/Intersection_\(mathématiques\)](https://fr.wikipedia.org/wiki/Intersection_(mathématiques))
- [https://fr.wikipedia.org/wiki/Union_\(mathématiques\)](https://fr.wikipedia.org/wiki/Union_(mathématiques))
- https://fr.wikipedia.org/wiki/Théorie_des_ensembles
- https://fr.wikipedia.org/wiki/Base_de_données



25. 60 minutes... Zut!

Anna et Benno font une randonnée avec leurs parents. En route, il faut qu'ils passent par un tunnel. L'expérience leur a appris que chaque membre du groupe a besoin d'un laps de temps différent pour le traverser : Anna a besoin de 10 minutes tandis que Benno ne met que 5 minutes. La mère, elle, a besoin de 20 minutes et le père a besoin de 25 minutes.

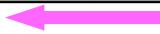
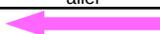
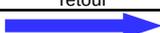


Le tunnel est sombre et très étroit, on doit le traverser soit seul soit à deux. Donc, pour que chaque membre du groupe arrive à l'autre bout du tunnel, ils doivent organiser leurs passages. Si deux personnes passent par le tunnel ensemble, le temps dont ils ont besoin correspondra au temps dont la personne la plus lente des deux aurait besoin pour le traverser seule. En plus, dans le tunnel, il faut que l'on se serve d'une lampe de poche.

Au moment où ils arrivent à l'entrée du tunnel, ils réalisent que la charge de la batterie de la lampe ne permet de l'utiliser que pendant 60 minutes. D'après toi, est-il possible que tous les membres du groupe arrivent à passer le tunnel en 60 minutes ?

Anna prétend que oui : «C'est tout à fait possible», et elle ajoute : «Pour passer par le tunnel, il nous faut cinq passages!»

Déplace les noms dans les champs correspondants pour que le plan d'Anna puisse être mis en œuvre.

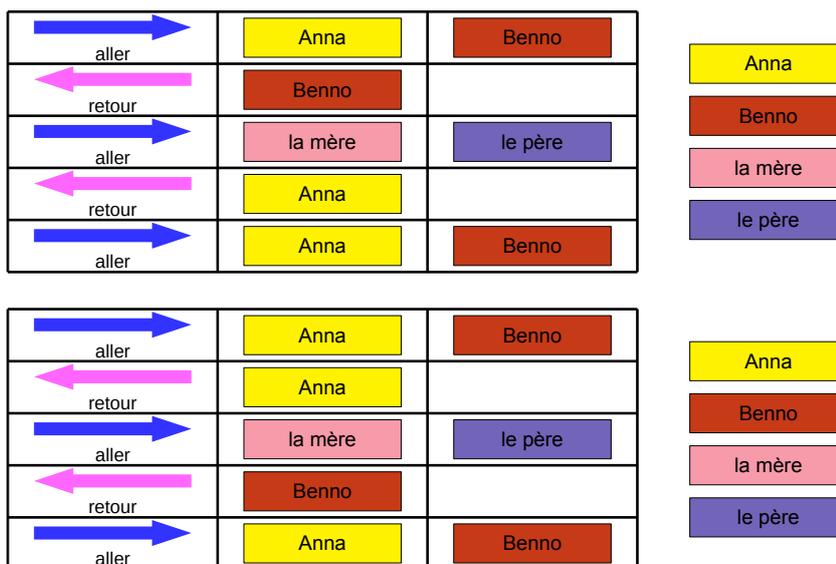
 aller		
 retour		
 aller		
 retour		
 aller		

Anna
Benno
la mère
le père



Solution

Pour que tous puissent passer par le tunnel en 60 minutes au maximum, il faut que la mère et le père, soit les deux membres du groupe les plus lents, ne le traversent qu'une seule fois. Ceci n'est possible que si Anna et Benno traversent le tunnel en premier (passage 1) et que l'un des deux retourne avec la lampe de poche (passage 2). Ainsi, celui-ci peut la rendre à ses parents qui passeront ensuite par le tunnel à deux (passage 3). Celui qui les attend à la sortie du tunnel reprend la lampe et retourne une dernière fois (passage 4) pour effectuer ensuite le passage final à deux. Peu importe si c'est Anna ou Benno qui effectueront le passage 2, le passage 4 sera effectué par l'un ou par l'autre, ce qui reviendra à un laps de temps de 15 minutes au maximum.



C'est de l'informatique !

Dans le domaine de l'informatique, on a souvent très peu de *ressources* pour résoudre des problèmes. En même temps, la solution d'un problème doit également répondre à certaines *contraintes secondaires*. Ainsi, pour résoudre un problème, il faut qu'on établisse un plan qui ne dépassera pas les ressources et qui répondra à toutes les contraintes secondaires.

Pour faire ceci, on prend souvent recours aux algorithmes d'*ordonnancement (scheduling)*. Comme les problèmes apparus dans la pratique se révèlent souvent très compliqués et que l'ordinateur aurait besoin de beaucoup de temps (voire probablement de plusieurs années) afin de trouver la solution la plus favorable, on se contente d'en trouver une qui mènera approximativement au résultat souhaité. Dans notre exemple, une telle solution est tout à fait convenable : on ne demande pas de trouver une solution pour que les membres du groupe puissent traverser le tunnel le plus vite possible. L'objectif est de ne pas dépasser les 60 minutes, c'est-à-dire, une solution qui réponde à la contrainte qu'on ne puisse pas passer par le tunnel sans lampe. Quand on a trouvé la solution qui répond convenablement à cette contrainte, on a résolu le problème.

Sites web et mots clés

ordonnancement, contraintes secondaires, ressources

— https://fr.wikipedia.org/wiki/Ordonnancement_de_tâches_informatiques



26. Le code KIX

Aux Pays-Bas, les codes postaux se composent de quatre caractères. Ces-derniers peuvent être des lettres ou des chiffres. Il y a même un code-barres propre aux codes postaux appelé code KIX. Chaque symbole du code KIX est subdivisé en deux parties : une partie supérieure (deux barres longues et deux barres courtes) et une partie inférieure (également deux barres longues et deux barres courtes). Dans la zone centrale du symbole, les deux barres courtes se superposent. Le tableau montre le code KIX composé pour les chiffres 0, 7, G, Y :

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

Le code KIX du code postal G7Y0 est donc :

À quel code postal le code KIX correspond-il ?



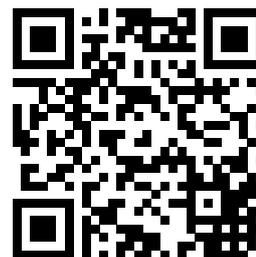
Solution

La réponse correcte est BC16 :

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

C'est de l'informatique !

Aux Pays-Bas, on applique en effet le code KIX pour les codes postaux. Des codes sous forme lisible par machine permettent, par exemple, de trier des lettres et des colis de manière automatique. De tels codes sont en outre souvent utilisés dans la vie quotidienne, par exemple, sous forme de codes à barres («bar» en anglais comme «barre» en français) qui sont décodés par des lecteurs de code-barres des caisses des supermarchés ou bien sous forme de code QR (pour «quick response») qui a été inventé de l'industrie automobile pour étiqueter des pièces automobiles. Entre-temps, ces types de codes, notamment le code QR, trouvent leur application surtout dans la publicité et il existe beaucoup d'applications pour le smartphone qui permettent leur décodage rapide. Dépêche-toi, scanne ce code QR pour décodé son contenu!



Sites web et mots clés

code KIX, code à barres, code QR

- <https://nl.wikipedia.org/wiki/KIX-code>
- <https://fr.wikipedia.org/wiki/Code-barres>
- <https://fr.wikipedia.org/wiki/RM4SCC>



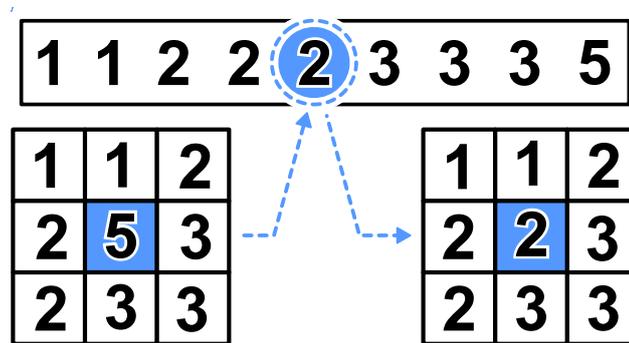
27. Le filtre médian

Une image est sauvegardée comme un tableau indiquant les valeurs de luminosité (entre 1 et 5) pour chaque pixel. La valeur 1 correspond à la couleur noire, la valeur 5 à la couleur blanche. Les valeurs 2 à 4 correspondent aux tons du gris de plus en plus clair.

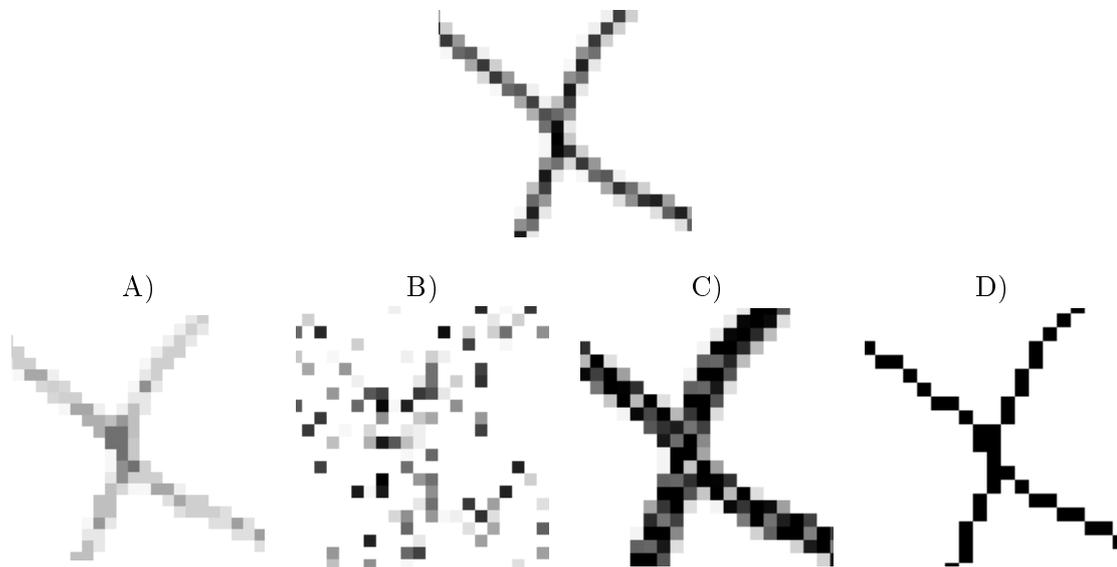
Un filtre appelé «filtre médian» influence le ton du gris de chaque pixel de l'image de sorte que...

- ... la valeur du pixel ainsi que les valeurs avoisinantes seront notées selon une séquence ordonnée croissante...
- ... le pixel assumera la cinquième valeur, soit la valeur médiane de la séquence, en tant que nouveau ton gris.

L'exemple ci-dessous montre que le pixel qui se trouve au centre passe de la valeur 5 à la valeur 2 :



Quel aspect cette image aura-t-elle au moment où on aura appliqué le filtre médian ?





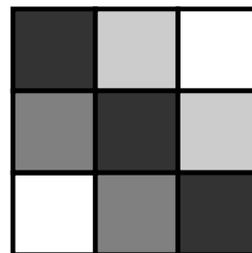
Solution

La réponse A) est correcte :

À travers le filtre médian choisi, des fragments d'image carrés sont traités avec neuf pixels. Le pixel noir au centre obtient une nouvelle valeur. Quand on regarde l'exemple à droite de plus près, on se rend compte qu'il n'y a que trois pixels qui sont noirs. Le médian de la séquence des tons gris filtrés est donc plus clair que l'original non filtré. Il s'ensuit que le noir changera en un ton gris.

Sur l'image originale de notre tâche, on voit que dans tous les fragments carrés à 9 pixels, les pixels noirs sont déjà minoritaires. C'est la raison pour laquelle l'image où on a appliqué le filtre médian ne comporte plus de pixels noirs.

Ceci n'est le cas que dans l'image A).



C'est de l'informatique !

Dans le traitement des photos, on aimerait bien réaliser rapidement certains effets visuels comme par exemple une plus grande netteté ou des couleurs plus lumineuses. Parfois, on aimerait bien obtenir des effets artistiques afin de personnaliser une photo. Ce sont justement de tels effets que l'on peut réaliser à l'aide des filtres d'image.

Le filtre médian fait partie de tels filtres. On applique le filtre médian pour corriger, par exemple, des erreurs de pixel qui se sont produites à cause d'un défaut du capteur d'images. Il en résultera donc que l'image semblera comme lissée et que chaque pixel sera uniformisé. On arrive ainsi aussi à réduire ou à atténuer certaines formes de bruits.

Sites web et mots clés

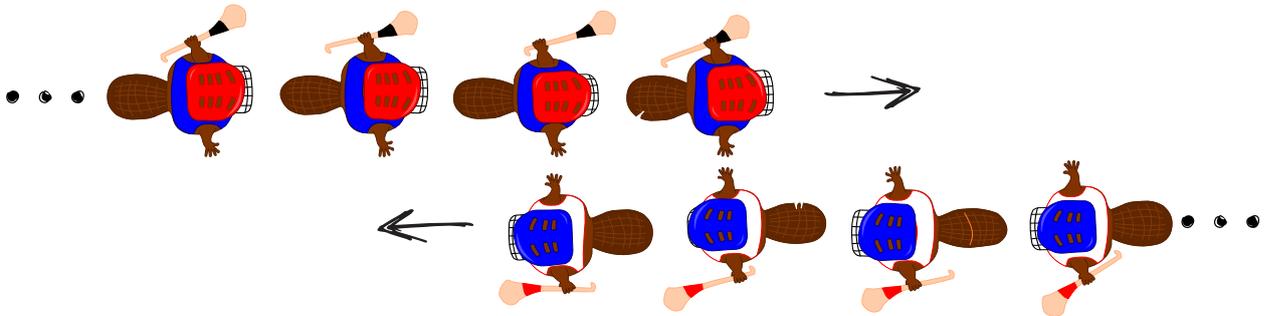
traitement des images, filtre médian, nuances de gris

— https://fr.wikipedia.org/wiki/Filtre_médian



28. La poignée de main style zip

Les castors adorent jouer au jeu irlandais appelé «Hurling». À la fin d'une partie, les joueurs des deux équipes se mettent en rang, l'un derrière l'autre. Ensuite, les deux équipes se croisent, toujours en rang, et les joueurs se serrent mutuellement la main pour remercier chaque joueur du jeu.



Plus exactement, ce rituel se déroule comme suit : D'abord, les deux premiers joueurs des deux équipes se serrent la main. Ensuite, les deux premiers joueurs serrent la main du deuxième joueur de l'équipe adverse (voir image ci-dessus). Et ainsi de suite, jusqu'à ce que les deux derniers joueurs des deux équipes se soient serré la main.

Le «Hurling» exige 15 joueurs par équipe. Le temps dont chaque joueur a besoin pour serrer la main du joueur de l'équipe adverse et pour avancer vers le prochain joueur est à une seconde.

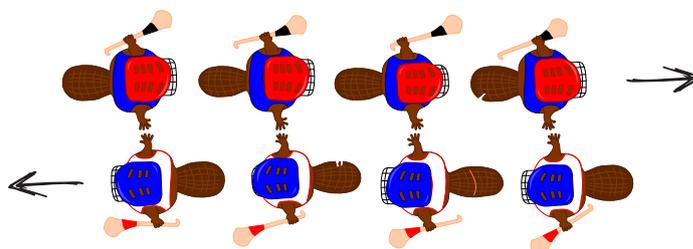
Combien de secondes faut-il au total pour que les deux équipes aient terminé le rituel de la poignée de main style zip ?



Solution

La réponse correcte est 29.

Le rituel de la poignée de main se déroule comme suit : le premier joueur de chaque équipe serre la main de chaque joueur de l'équipe adverse. Après quatre joueurs, on sera à quatre secondes et la situation se présentera ainsi :



En même temps, le dernier joueur de chaque équipe serre d'abord la main à un joueur de l'équipe adverse. Ensuite, il doit serrer la main à tous les autres joueurs. Si on suppose quatre joueurs, ceci prendra trois secondes de plus.

Il est donc possible de généraliser et d'affirmer que si on suppose n joueurs, la durée peut être calculée comme suit : d'abord on a besoin de n secondes, ensuite encore une fois de $n - 1$ secondes. Finalement, on arrive à l'équation $n + n - 1 = 2n - 1$ secondes. Dans le cas des 15 joueurs, le rituel de la poignée de main style zip durera donc $15 + 15 - 1 = 29$ secondes.

C'est de l'informatique !

Pour les équipes du jeu Hurling avec 15 joueurs, nous avons donc été capables de calculer exactement la durée de la poignée de main. On imagine que les spectateurs arrivent à bien supporter d'attendre ces 29 secondes jusqu'à ce que ce rituel soit terminé. Cependant, prenons une équipe de hockey qui présente 22 joueurs ? Peut-on toujours recourir à l'algorithme appliqué aux équipes Hurling ou est-ce que cette procédure durera trop longtemps ? Ce serait en effet très utile de disposer d'un algorithme capable d'évaluer le temps d'exécution sans devoir tout calculer à nouveau en détail.

L'informatique se consacre à une recherche intensive des algorithmes capables d'analyser le temps d'exécution. Ces analyses fournissent une expression mathématique comprenant une variable n qui représente la taille des données saisies. Dans le cas de la poignée de main du jeu Hurling, nous obtenons une telle expression si nous remplaçons la deuxième phrase de la réponse : on remplace par n la partie de la phrase «le nombre des joueurs d'une équipe» (15) par $2n - 1$. Il est ainsi possible de calculer exactement le temps d'exécution aussi pour d'autres nombres de joueurs : pour 22 joueurs, par exemple, le temps d'exécution sera à 43 secondes ($2n - 1 = 2 \cdot 22 - 1 = 43$ secondes), pour 40 joueurs, le temps d'exécution sera à 79 secondes et ainsi de suite.

L'expression du temps d'exécution $2n - 1$ est en fait une fonction linéaire. L'algorithme de «la poignée de main» appartient donc à la classe des algorithmes à fonction linéaire appelée aussi $O(n)$. Imaginons encore, si on serrait la main d'une autre manière. Quelles en seraient les conséquences, par exemple, si chacun serrait la main de l'autre individuellement (et non en même temps) ? Dans ce cas-là, l'algorithme appartiendrait à la classe $O(n^2)$ et les deux équipes du jeu Hurling se serreraient la main mutuellement pendant $15^2 = 225$ secondes, soit environ quatre minutes. Si l'algorithme avait un temps d'exécution exponentiel, il se trouverait dans la classe $O(2^n)$ et les pauvres joueurs devraient se serrer la main mutuellement pendant environ $2^{15} = 32768$ secondes, ce qui reviendrait à un temps d'exécution de 9 heures ! Les spectateurs, eux, ils auraient sans aucun doute quitté les rangs depuis longtemps pour aller se coucher... Il s'avère donc judicieux de réfléchir au fait de pouvoir exécuter quelques opérations en même temps pour épargner un peu de temps.



Sites web et mots clés

complexité du temps d'exécution, analyse du temps d'exécution

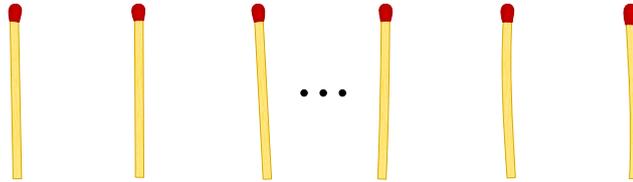
— https://fr.wikipedia.org/wiki/Comparaison_asymptotique





29. Les jeux de Nim

Beat et son ami jouent aux jeux de Nim. Sur la table, il y a 13 allumettes. À tour de rôle, les deux joueurs enlèvent soit 1 soit 2 soit 3 allumettes. Le joueur qui enlève la dernière allumette a gagné la partie.



Remarque : S'il n'y a que quatre allumettes sur la table, Beat ne peut plus gagner. C'est exactement cette situation qu'il vise à éviter.

C'est le tour de Beat. Combien d'allumettes doit-il enlever pour gagner cette partie ?

- A) 1
- B) 2
- C) 3
- D) Peu importe combien.



Solution

La réponse A) est correcte. Ainsi, il ne reste que 12 allumettes. Son ami peut maintenant enlever 1, 2 ou 3 allumettes et Beat en prend autant pour qu'il n'en reste que 8. Ensuite, c'est le tour de son ami qui a le choix d'enlever soit 1, 2 ou 3 allumettes. Finalement, Beat n'a qu'à en enlever autant pour qu'il n'en reste que 4. Son ami n'aura donc plus de chances de gagner la partie.

Si Beat enlevait 2 ou 3 allumettes au début de la partie, son ami enlèverait, en toute logique, autant d'allumettes pour qu'il n'en reste qu'un multiple du nombre 4 ... et Beat serait le perdant.

C'est de l'informatique !

Dans la théorie des jeux, les jeux comme les jeux de Nim ou le dilemme du prisonnier servent de modèle pour analyser des problèmes stratégiques réels afin d'en déduire une méthode de résolution. Dans le domaine de l'économie de marché, par exemple, les conclusions de ces jeux de réflexion peuvent servir à fixer les prix de façon optimale. D'une part, il est un fait bien connu que des baisses de prix peuvent augmenter les ventes. D'autre part, une telle décision peut affecter considérablement la marge bénéficiaire de chaque produit vendu. À l'inverse, une augmentation de prix permettra d'augmenter la marge bénéficiaire de chaque produit vendu. Pourtant, une telle décision peut affecter les ventes et, en conséquence logique, le bénéfice total de la société. Ainsi, les modèles appliqués dans la théorie des jeux peuvent aider à prédire de possibles réactions des acheteurs aux ajustements des prix. On ne comprend l'immense valeur des modèles de la théorie des jeux pour l'économie que quand on réalise qu'il existe déjà plusieurs travaux scientifiques sur des modèles de la théorie des jeux couronnés du Prix Nobel.

Sites web et mots clés

jeux de Nim, théorie des jeux, arbre de décision

- <http://www.mathematische-basteleien.de/nimspiel.html>
- <http://scienceblogs.de/zoonpolitikon/2008/04/22/spieltheorie-einfach-erklart-i-einleitung-und-gefangenendilemma/>

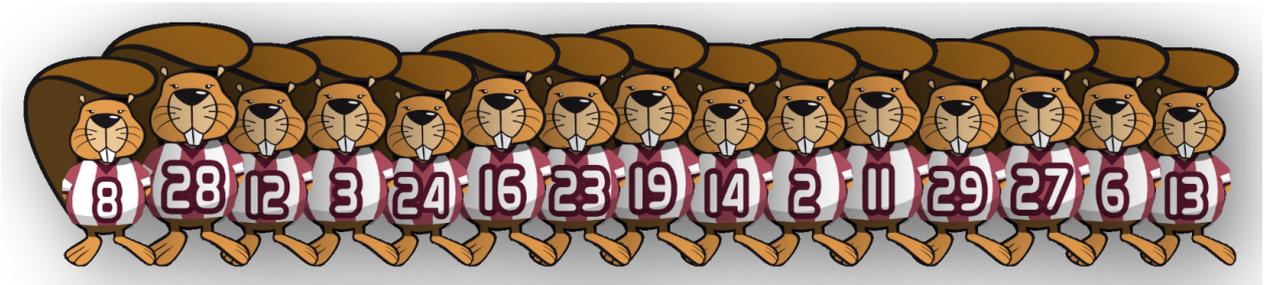
30. Le classement des numéros de maillot

Les images suivantes montrent deux équipes de 15 joueurs. Les joueurs des deux équipes portent des maillots imprimés avec un numéro individuel. Ceux de la première équipe se tiennent debout, l'un à côté de l'autre et sont classés par numéro. Ceux de la seconde équipe se tiennent également debout mais ils ne sont pas classés par numéro.

L'équipe no. 1 :



L'équipe no. 2 :



Comment peut-on savoir le plus vite possible quels numéros sont représentés tant dans l'équipe no. 1 que dans l'équipe no. 2 ?

- A) On parcourt les numéros de l'équipe no. 1 (1, 4, 5, ...) et on vérifie si ces numéros existent également dans l'équipe no. 2.
- B) On parcourt les numéros de l'équipe no. 2 (8, 28, 12, ...) et on vérifie si ces numéros existent également dans l'équipe no. 1.
- C) Peu importe avec quelle équipe vous commencez. La durée pour vérifier tous les nombres est la même pour les deux équipes.
- D) Il s'avère judicieux de vérifier d'abord combien de nombres n'existent pas dans les deux équipes. En soustrayant ce nombre du nombre total de 15, on obtient le nombre recherché.



Solution

La réponse correcte est B). Dans une série ordonnée, on peut déterminer une certaine valeur, comme par exemple les nombres, plus rapidement que dans une série non-ordonnée. Un ordinateur aurait besoin de seulement $\log_2(n)$ étapes de recherche au lieu de n étapes. Avec le procédé décrit dans la réponse B), il aurait donc besoin au pire des cas de $n \cdot \log_2(n)$ étapes de recherche.

Avec le procédé décrit dans la réponse A), on ne pourrait pas parcourir la liste des numéros de manière rapide. Bien au contraire, il faudrait que l'on parcoure tous les nombres un par un. Un tel procédé demanderait au pire des cas n^2 étapes de recherche, ce qui reviendra donc à plus de $n \cdot \log_2(n)$ étapes de recherche.

Par conséquent, la réponse C) ne peut pas être juste car n^2 n'est pas égal à $n \cdot \log_2(n)$.

Le procédé décrit dans la réponse D) n'aboutit en général pas au résultat souhaité. Voilà pourquoi cette réponse sera rejetée.

C'est de l'informatique !

Il est beaucoup plus facile de fouiller une série ordonnée qu'une série non-ordonnée. Regardons de plus près comment tu as cherché par exemple le numéro 9 dans l'équipe no. 1. D'abord, tu regardes le joueur situé au milieu de l'équipe qui porte le numéro 17. Le numéro recherché est plus petit que le numéro 17, voilà pourquoi tu continues à chercher le numéro 9 dans la partie gauche de la série ordonnée. Là, tu fixes d'abord la partie centrale. Tu y découvres le numéro 7. Finalement, tu regardes à droite et tu y découvres immédiatement le numéro 9. L'astuce est que tu divises la zone de recherche par deux à chaque étape de ta recherche et que, de cette manière, tu la réduis à une taille beaucoup plus transparente. En informatique on parle de «recherche binaire». Ce mot d'origine latine comprend le terme *bis* qui, lui, signifie «deux fois». Le classement prend une place importante en informatique car la recherche binaire permet des opérations beaucoup plus rapides à l'intérieur des ensembles de données ordonnées. Ainsi, dans le cas d'une recherche de n nombres, la recherche binaire n'aura besoin que d'environ $\log_2(n)$ étapes de recherche.

Sites web et mots clés

classement, recherche binaire

— https://fr.wikipedia.org/wiki/Recherche_dichotomique



31. Cassy, la tortue

La tortue appelée Cassy habite un pays dont le sol est quadrillé. Le champ sur lequel elle bouge est assez restreint : elle ne vit que sur une parcelle de cinq fois cinq cellules de grille. Elle adore plus que tout manger de la salade fraîche. Quelle chance : tous les matins, les salades poussent de nouveau sur le champ. Cassy ne sait jamais où elles repousseront exactement, cependant, ce qu'elle sait très bien c'est qu'elle veut les manger toutes ! Pour ce faire, Cassy démarre chaque matin au milieu du champ et suit les instructions selon le bloc d'instructions.

Procède en sorte que Cassy parcourt toutes les cellules de grille de son champ à salades.

- R est une variable de comptage. Quand le bloc d'instructions est exécuté la première fois R est égal à 1, à la deuxième exécution, R est égal à 2, et ainsi de suite.
- Cassy peut quitter le champ, mais elle n'est pas autorisée à quitter son pays.
- En cliquant sur le bouton «tester», tu peux tester ton programme.

The image shows a programming interface for a turtle named Cassy. On the left, there are three instruction blocks: "tourner à gauche" (turn left), "tourner à droite" (turn right), and "R cellules en avant" (move R cells forward). These three blocks are grouped together in a larger box labeled "instructions". This group is then placed inside a "bloc d'instructions" (instruction block) which is surrounded by a dashed line and labeled "À répéter cinq fois!" (Repeat five times!). Below the instruction block is a red button labeled "tester". To the right of the code is a 10x10 grid. The central 5x5 area of the grid is shaded brown, representing the field where Cassy lives. A small green turtle icon is positioned in the center of this 5x5 field.



Solution

Il est possible d'insérer quatre instructions au maximum par bloc d'instructions et chaque bloc est répété cinq fois. Ceci signifie que le chemin de la tortue ressemblera à une spirale. Il existe quatre séquences d'instructions que Cassy pourrait suivre afin de parcourir le champ entier et en spirale.

— R cellules en avant, tourner à gauche, R cellules en avant, tourner à gauche

À répéter cinq fois!

- R cellules en avant
- tourner à gauche
- R cellules en avant
- tourner à gauche

— R cellules en avant, tourner à droite, R cellules en avant, tourner à droite

À répéter cinq fois!

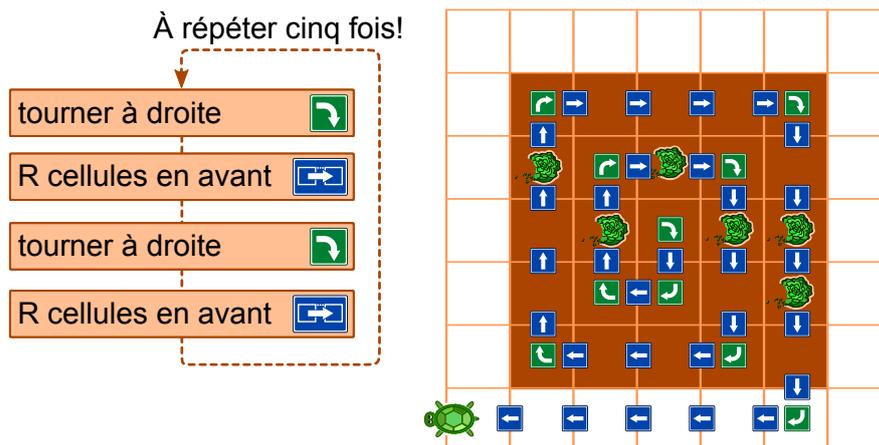
- R cellules en avant
- tourner à droite
- R cellules en avant
- tourner à droite

— tourner à gauche, R cellules en avant, tourner à gauche, R cellules en avant

À répéter cinq fois!

- tourner à gauche
- R cellules en avant
- tourner à gauche
- R cellules en avant

— tourner à droite, R cellules en avant, tourner à droite, R cellules en avant



C'est de l'informatique !

Cette tâche nécessite une programmation informatique. Le programme comportera une séquence d'instructions (un bloc d'instructions) qui sera répétée cinq fois. Ce type de répétition fait partie des répétitions cycliques rassemblées sous le terme «boucle informatique». Dans le cas présent, on parlera d'une boucle de comptage car R passe de 1 à 5. Le processeur d'un ordinateur exécute ces instructions dans l'ordre. Tous les langages de programmation comprennent des boucles – ainsi que d'autres instructions qui peuvent contrôler le déroulement du programme, comme par exemple les branchements et les appels aux sous-programmes.

Si le programme fonctionne correctement, l'ordinateur se comportera exactement comme tu l'avais prévu. Si le programme ne fonctionne pas correctement, l'ordinateur procédera à l'exécution des instructions. Pourtant, Cassy la tortue ne parcourra pas les champs comme tu l'avais prévu. Habituellement, un ordinateur n'est pas capable de reconnaître si un programme créé et exécuté aboutira à ses – disons à tes – fins !

Sites web et mots clés

la fenêtre graphique Turtle

- <http://www.turtlegrafik.ch/>
- <http://primalogo.ch/fr>

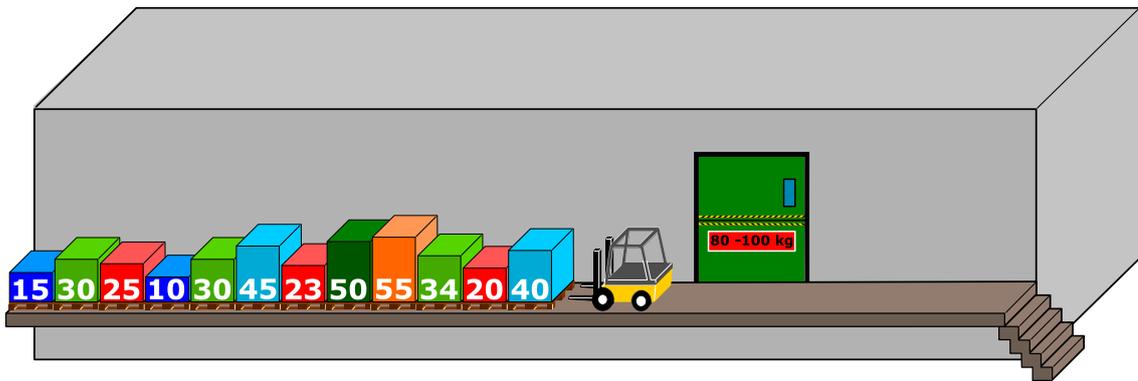




32. Chaque palette en son temps !

Sur une plate-forme de chargement très étroite, il y a des palettes placées l'une à côté de l'autre. Chaque palette présente un poids différent (de gauche à droite) : 15 kg, 30 kg, 25 kg, 10 kg, 30 kg, 45 kg, 23 kg, 50 kg, 55 kg, 34 kg, 20 kg, 40 kg. La plate-forme est étroite au point que l'on ne peut pas déplacer les palettes en les soulevant l'une au-dessus de l'autre.

Afin de transporter les palettes dans un entrepôt, il faut les mettre dans un ascenseur de charge. Cet ascenseur transporte les palettes à l'entrepôt quand la charge comporte au moins 80 kg. Pourtant, la charge ne doit pas dépasser les 100 kg. Au moment où l'ascenseur est déchargé, il repart vide vers le niveau de la plate-forme.



L'élévateur de palettes déplace généralement d'abord la palette qui se trouve le plus près du monte-charge. Au cas où le poids total dépasse les 100 kg lors du chargement de l'ascenseur avec la dernière palette, l'élévateur de palettes dépose cette dernière à l'autre bout de la plate-forme (à droite). Si le poids total ne dépasse pas les 100 kg, la palette restera dans le monte-charge.

Quand toutes les palettes du côté gauche de la plate-forme se trouveront dans l'entrepôt, l'élévateur de palettes déplacera de la même manière toutes les autres palettes du côté droit de la plate-forme afin de les déposer dans le monte-charge.

Laquelle des affirmations ci-dessous est correcte ?

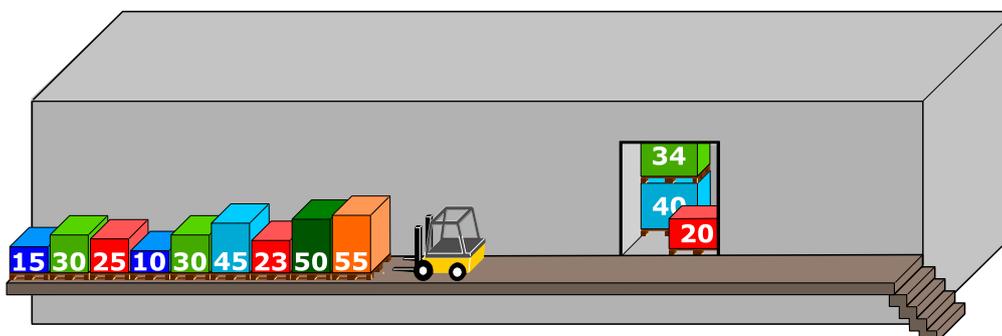
- A) La deuxième charge de l'ascenseur s'élève à 98 kg.
- B) On n'utilise pas les palettes de l'autre bout de la plate-forme.
- C) Une charge du monte-charge s'élève à 100 kg.
- D) Le monte-charge descend et monte cinq fois au total.
- E) La méthode pour déplacer les palettes décrite ci-dessus ne permet pas de transporter les palettes dans l'entrepôt.



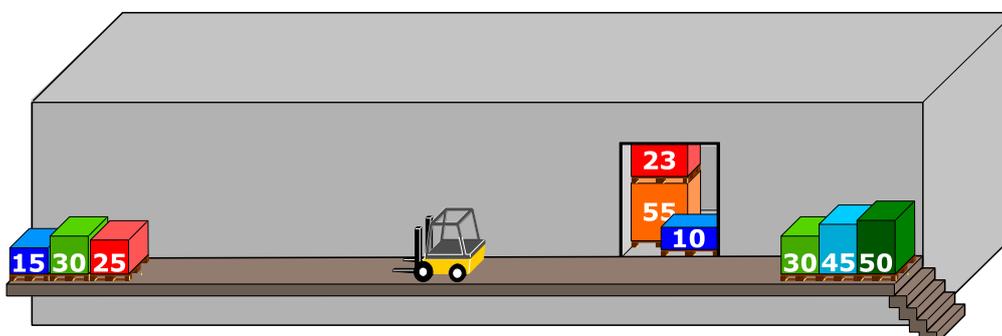
Solution

La réponse correcte est C) :

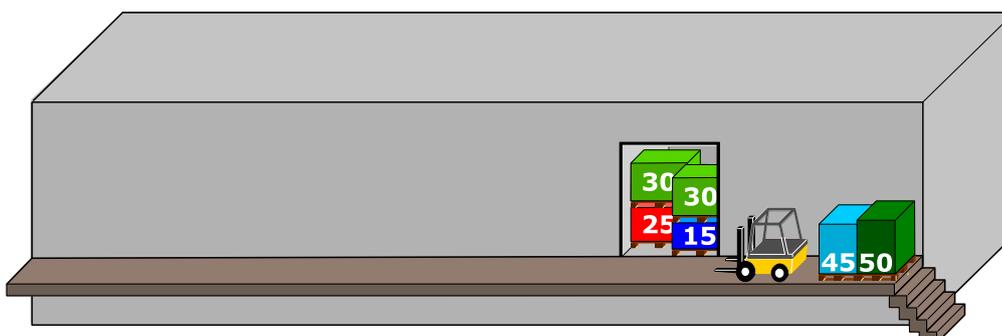
- La première charge comporte les trois premières palettes. Leur poids s'élève à : $40 \text{ kg} + 20 \text{ kg} + 34 \text{ kg} = 94 \text{ kg}$.



- La deuxième charge ne comporte qu'une seule palette (55 kg). Si on y ajoutait la palette suivante (50 kg), la charge dépasserait les 100 kg, c'est la raison pour laquelle l'élévateur de palettes la dépose au côté droit de la plate-forme de chargement. Ensuite, il charge l'ascenseur avec la palette suivante (23 kg) ; cette charge sera encore trop légère : le poids total ne s'élèvera qu'à $55 \text{ kg} + 23 \text{ kg} = 78 \text{ kg}$. Avec la palette suivante (45 kg), par contre, la charge dépasserait de nouveau le poids maximal de 100 kg. Voilà pourquoi l'élévateur de palettes déposera la palette de 45 kg au côté droit de la plate-forme, juste à côté de la palette à 50 kg. On procédera de la même manière avec la palette suivante (30 kg). Finalement, il est possible de charger l'ascenseur avec la palette dont le poids s'élève à 10 kg seulement. Ainsi, l'ascenseur repartira vers l'entrepôt avec la charge de $55 \text{ kg} + 23 \text{ kg} + 10 \text{ kg} = 88 \text{ kg}$.



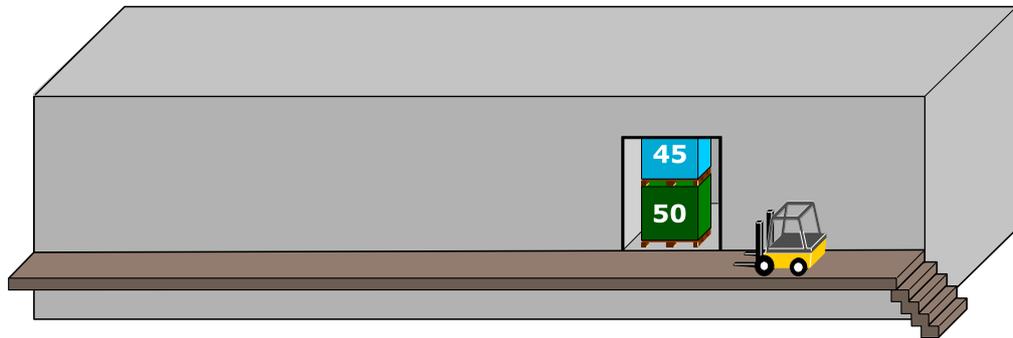
- L'élévateur de palettes déplace ensuite les trois dernières palettes ($25 \text{ kg} + 30 \text{ kg} + 15 \text{ kg} = 70 \text{ kg}$) et les dépose dans le monte-charge. Il ajoutera à cette troisième charge la palette de 30 kg se trouvant actuellement comme première dans la rangée des palettes au côté droit de la plate-forme. Ainsi, la charge de l'ascenseur s'élèvera exactement à 100 kg.



- Finalement, il ne restent que les deux palettes du côté gauche de la plate-forme :



45 kg + 50 kg = 95 kg. Elles constitueront la quatrième et dernière charge que l'élevateur de palette déposera dans le monte-charge.



Prenant en compte les explications ci-dessus, les réponses A), B), D) et E) ne sont pas correctes.

C'est de l'informatique !

Comme la plate-forme est si étroite, l'élevateur ne peut déplacer que la première palette de la rangée avant de traiter les suivantes. Ce traitement est comparable au principe de l'empilage : on ne peut enlever que l'objet qui se trouve tout en haut de la pile et, de manière analogue, on ne peut ajouter un autre objet qu'en le déposant sur l'objet qui se trouve tout en haut de la pile. En informatique, on nomme une telle structure de données également «une pile». Dans la présente tâche, nous avons trois piles : la rangée de palettes de gauche (avec un premier élément à l'extrême droite), la rangée de palettes à droite (avec un premier élément à l'extrême gauche) et le monte-charge.

Au quotidien, nous connaissons aussi des piles : quand un enfant construit une tour avec des blocs, il ne peut enlever ou ajouter un bloc si ce-dernier se trouve tout en haut de la pile. Il en est de même pour un cornet de glace à plusieurs boules : la boule ajoutée en dernier sera mangée en premier. Le même principe s'applique à la navigation sur Internet, car la sauvegarde des pages visitées sur Internet suit le principe de l'empilage : en cliquant sur «retour», la page visitée en dernier sera affichée en premier.

Ce principe est également appelé LIFO : „last in – first out“ («dernier arrivé, premier sorti»). Il s'oppose au principe FIFO : „first in – first out“ («premier arrivé, premier sorti»). Le principe FIFO est un mode de traitement que l'on applique dans le cas d'une queue au guichet ou dans un cabinet médical (bien que les cas d'urgence puissent heureusement être traités en priorité).

Sites web et mots clés

structure de données, pile, algorithme, principe LIFO

- [https://fr.wikipedia.org/wiki/File_\(structure_de_donn%C3%A9es\)](https://fr.wikipedia.org/wiki/File_(structure_de_donn%C3%A9es))
- https://fr.wikipedia.org/wiki/Last_in,_first_out





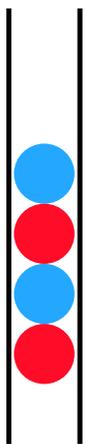
33. Jeux de boule

Emil s’amuse à jouer à un nouveau jeu vidéo sur son ordinateur. Le jeu démarre avec une pile d’au moins trois boules colorées (en rouge ou en bleu). Ces boules se trouvent à l’intérieur d’un tube. En cliquant sur un bouton, les deux boules qui se trouvent en bas de la pile sont poussées en avant et tombent du tube. En plus, au moment où elles tombent, de nouvelles boules tombent d’en haut pour se rajouter à la pile de boules dans le tube. En fonction de la couleur des boules qui se trouvent en bas de la pile, ce processus peut avoir deux conséquences :

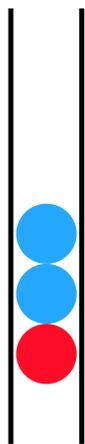
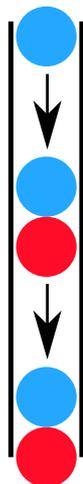
Si la boule qui se trouve en bas de la pile est rouge, une boule bleue tombe d’en haut et rejoint les boules empilées dans le tube :

Si la boule qui se trouve en bas de la pile est bleue, trois boules aux couleurs rouge, bleu et rouge rejoignent les boules empilées dans le tube :

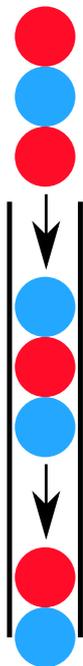
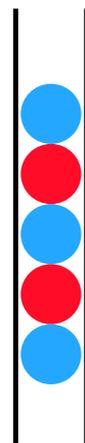
avant



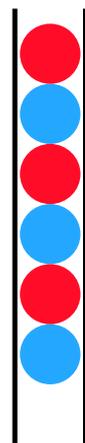
après



avant



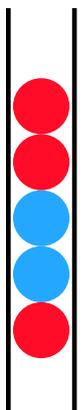
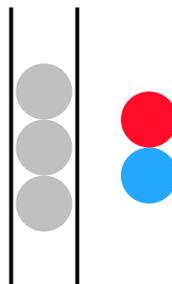
après



Emil répète le clic sur le bouton jusqu’à ce qu’il reste au moins trois boules dans le tube. Le jeu se termine au moment où il y a moins de trois boules dans le tube.

Si Emil commence avec une pile comme elle est représentée dans la figure de droite, le jeu sera bientôt terminé car après cinq clics, il ne restera que deux boules dans le tube.

Choisis des boules colorées de la pile de droite pour les insérer dans les places d’empilement libres afin que tu aies une pile de départ qui permettra un jeu infini.

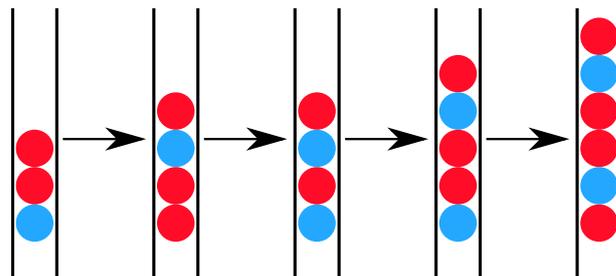
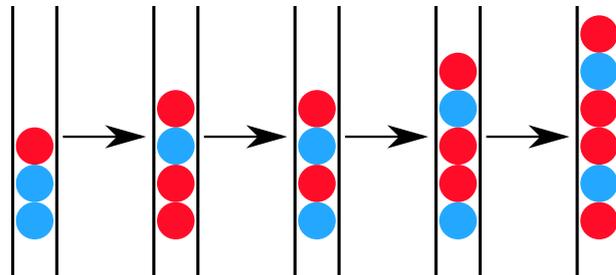
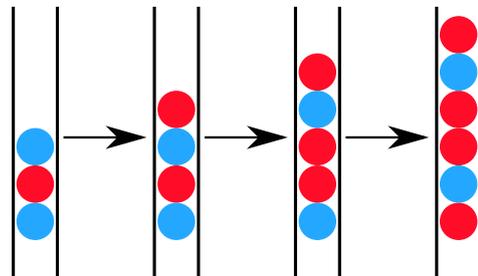
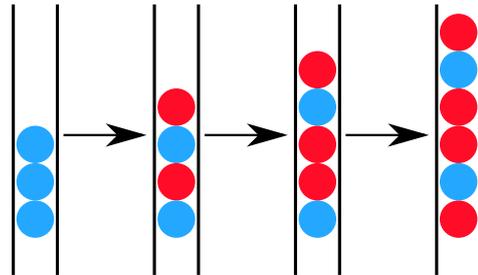




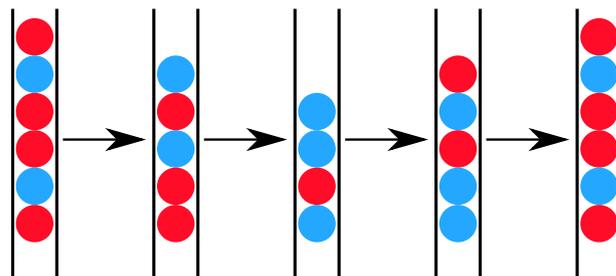
Solution

Si la boule qui se trouve tout en bas de la pile est rouge, le jeu se terminera tout de suite après le premier clic sur le bouton parce que après ce premier clic, il ne restera que deux boules dans le tube.

Mais, si la boule qui se trouve tout en bas de la pile est bleue, le jeu continuera à l'infini. Avec au maximum quatre clics sur le bouton, on obtient de nouveau la pile à six boules avec la séquence rouge-bleu-rouge-rouge-bleu-rouge :



Par la suite, le jeu continue selon un cycle à quatre étapes en tant que cycle infini :





C'est de l'informatique !

Le jeu présenté dans cette tâche a servi comme exemple lors d'une démonstration de la part d'Emil Leon Post. Avec cet exemple, il voulait montrer qu'il est possible de créer des processus infinis durant la réécriture des chaînes de caractères («strings»). Emil Leon Post (1897-1954) était un mathématicien et logicien polonais qui a publié beaucoup d'articles relatifs à l'informatique théorique notamment dans le domaine de la logique propositionnelle. On peut décrire un tel modèle de réécriture à l'aide de la grammaire formelle. À partir de l'alphabet d'entrée, on définit les règles selon lesquelles on substituera des caractères. Dans notre exemple, la grammaire connaîtra quatre règles (où X représentera une quelconque combinaison de boules bleues et rouges) :

$$Xbb \rightarrow rbrX$$

$$Xrb \rightarrow rbrX$$

$$Xbr \rightarrow bX$$

$$Xrr \rightarrow bX$$

Sites web et mots clés

modèle de calcul, langue formelle, règle de production, chaîne de caractères, système de traitement de données

- http://esolangs.org/wiki/Post_canonical_system
- https://en.wikipedia.org/wiki/Tag_system
- https://en.wikipedia.org/wiki/Post_canonical_system



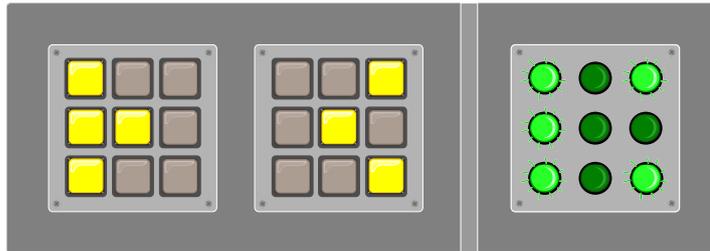


34. Deux possibilités exclusives

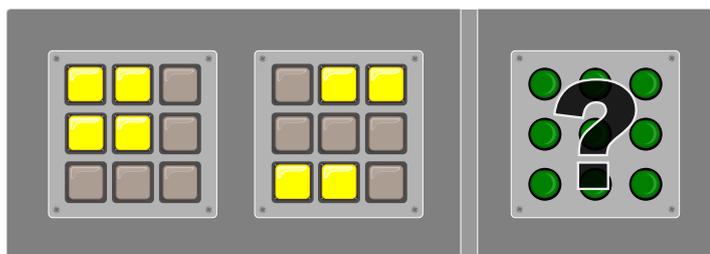
Erich a trouvé un vieil appareil électronique. Sur le côté gauche, il y a deux champs et chacun dispose de 9 touches sur lesquelles on peut appuyer. Sur le côté droit, il y a un champ avec 9 lampes. En fonction de la combinaison des touches, les lampes s'allument ou s'éteignent.

Erich découvre que la position d'une lampe qui s'allume et s'éteint correspond à une combinaison de touches particulière dans les deux champs.

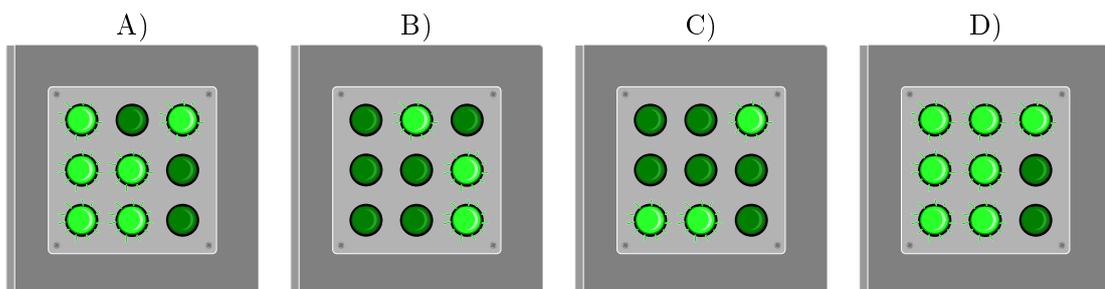
Actuellement, les lampes sont allumées comme suit :



Maintenant, Erich a envie de changer la combinaison des touches. Voilà le résultat :



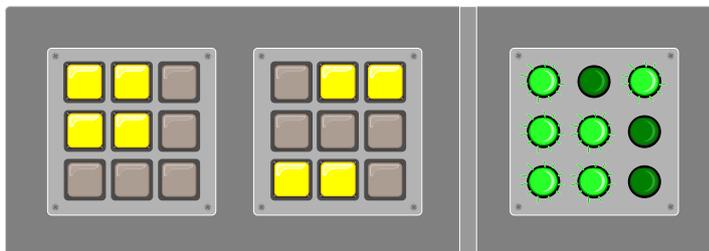
Lesquelles des lampes sont allumées ?





Solution

La réponse correcte est A). La règle est : si les touches de la même position dans les deux champs à gauche ne sont allumées ou éteintes qu'une seule fois, la lampe correspondante est allumée, sinon elle est éteinte.



C'est de l'informatique !

Dans tous les domaines de l'informatique, on a volontiers recours à la logique binaire pour vérifier dans une situation d'application particulière si une affirmation est vraie ou fausse. Ceci n'est possible que s'il n'existe que deux possibilités exclusives (deux valeurs). Une troisième possibilité sera ainsi exclue («*tertium non datur*»). Les fonctions logiques peuvent donc fournir deux valeurs logiques : soit «faux» soit «vrai».

Dans notre exemple, nous prenons recours à la fonction XOR (de l'anglais : eXclusive OR, «ou exclusif»). XOR fonctionne comme suit : partant de deux affirmations dont une seule est exclusivement «vraie», XOR livre la valeur logique «vrai». Pour tous les autres cas, XOR livre la valeur logique «faux». Dans l'exemple de notre tâche, la fonction logique XOR est appliquée neuf fois (touches) et elle livre deux affirmations (lampes).

Sites web et mots clés

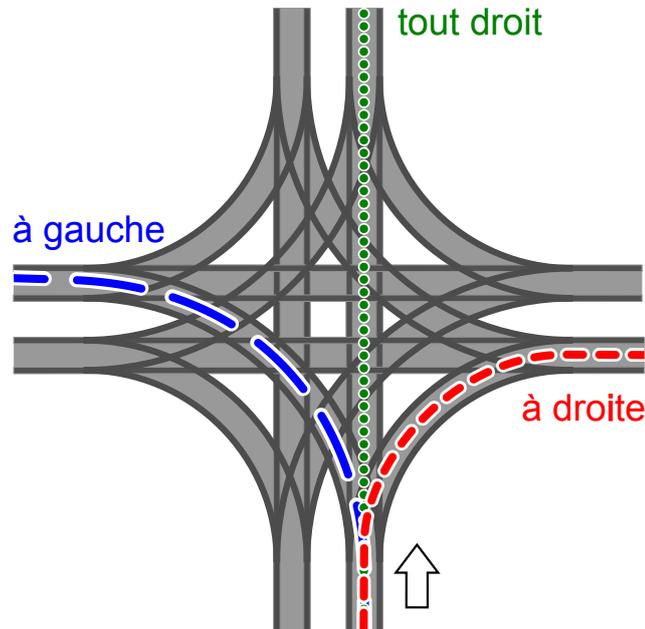
logique, binaire, fonction XOR

- https://fr.wikipedia.org/wiki/Fonction_OU_exclusif
- https://en.wikipedia.org/wiki/XOR_gate



35. Le carrefour des trams

Dans les rues de Saint-Petersbourg circulent des trams. Au centre-ville, il y a un carrefour qui leur permet de continuer dans toutes les directions : Quelle que soit la direction d'où ils arrivent, ils peuvent tourner à gauche, à droite ou continuer tout droit.



La direction dans laquelle les trams partent est déterminée par la position des aiguilles. Ces positions sont décrites à l'aide de la combinaison des mots suivants : TOUT DROIT, À GAUCHE, À DROITE. La combinaison À GAUCHE-TOUT DROIT-À GAUCHE-À DROITE, par exemple, signifie que la position des aiguilles est telle que le premier tram qui arrive tourne à gauche, le suivant (en sens horaire) continue tout droit, le succédant (toujours en sens horaire) tourne à gauche et le quatrième tram tourne à droite.

Il est également possible que les trams arrivent en même temps au carrefour tout en provenant de toutes les directions possibles.

Quelles positions des aiguilles risquent de provoquer une collision entre les trams ?

- A) À DROITE-À DROITE-À DROITE-À DROITE
- B) À DROITE-À DROITE-À GAUCHE- À GAUCHE
- C) À GAUCHE-À DROITE-À GAUCHE-À DROITE
- D) À DROITE-À GAUCHE- À DROITE- À GAUCHE



Solution

La réponse correcte est B). Le premier tram (qui tourne à droite) heurtera le troisième tram (qui tourne à gauche). De même, le deuxième tram (qui tourne à droite) heurtera le quatrième tram (qui tourne à gauche).

Toutes les autres solutions permettront aux trams de passer sans aucun incident : dans la solution A), tous les trams se trouvent sur la chaussée à l'extrême droite, dans la solution C) et D), il y a deux trams qui passent tranquillement l'un à côté de l'autre. Ici, il est important de noter que les chaussées des trams au centre ne se touchent pas. On appelle une chaussée où deux trams tournent à gauche sur deux chaussées séparées et parallèles une chaussée «tangentielle» ou «américaine».

C'est de l'informatique !

Les chaussées des trams – surtout en matière de croisements – représentent en fait des ressources auxquelles de différents acteurs ont recours et qu'ils partagent conjointement. Dans notre exemple, afin d'éviter une collision, il est pourtant important qu'il n'y ait qu'un seul acteur (dans notre cas il s'agit d'un seul tram) par endroit. En informatique, la coordination et l'optimisation du partage des ressources de différents processeurs en cours est un domaine important. Tout comme dans le cas de la chaussée des trams, la technologie digitale requiert que l'on se rende compte des risques et que l'on respecte des règles qui garantissent par exemple l'accès sécurisé aux bases de données ou bien aux ordinateurs publics ou partagés. Dans le cas contraire, on risque des pertes de données ou la destruction matérielle de composants informatiques.

Dans le cas du carrefour des trams, il est judicieux de n'approuver que des positions d'aiguilles qui permettent un passage sécurisé des trams. Une solution tout à fait valable serait d'installer des signaux lumineux. L'informatique, elle aussi, connaît des solutions qui ont recours à des signaux appelés «sémaphores» (de l'anglais pour «signal»). Leur fonction est d'indiquer au processeur quelles ressources sont en train d'être utilisées afin qu'il attende son propre tour.

Sites web et mots clés

carrefour des trams, synchronisation des processus, accès aux ressources restreintes

- [https://fr.wikipedia.org/wiki/Synchronisation_\(multitâches\)](https://fr.wikipedia.org/wiki/Synchronisation_(multitâches))
- <http://www.swisseduc.ch/informatik/infotraffic/logictraffic/>

36. Le codage des images numériques

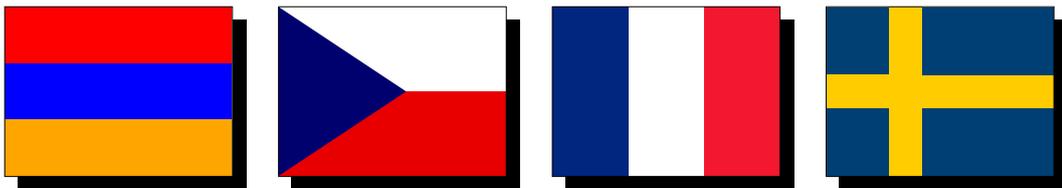
Les images numériques sont constituées d'un ensemble de points appelés « pixels ». Quand on veut enregistrer des images numériques en tant que données, dans le cas le plus simple, on décrit la couleur de chaque pixel individuellement. Le format de fichier GIW (imaginaire) permet de compresser les images numériques, c'est-à-dire de les enregistrer avec une taille réduite. Voilà comment ce procédé fonctionne :

- On code chaque ligne de pixels séparément.
- On code chaque couleur en lui attribuant une abréviation comprenant trois lettres.
- Une suite de pixels de la même couleur est codée par une paire de parenthèses comprenant l'abréviation des couleurs correspondantes ainsi que le nombre des pixels de la même couleur.

Exemple : la ligne de pixels codée par les paires de parenthèses (ver, 20) (blc, 13) comprend d'abord 20 pixels verts, puis 13 pixels blancs.

Voici quatre images numérique représentant quatre drapeaux. Le nombre de lignes de pixels est égal pour chaque image tout comme le nombre de pixels. Pour enregistrer les images, on choisit le format GIW.

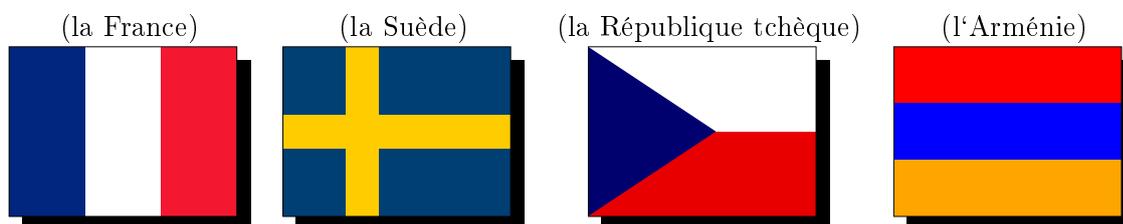
Classe les images selon la taille de leur fichier GIW !





Solution

Le bon ordre est le suivant :



Au cas où une ligne de pixels entière ne comporte qu'une seule couleur, il ne faut la coder que par une paire de parenthèses. Chaque changement de couleur entraîne une nouvelle paire de parenthèses. Dans le drapeau de l'Arménie, par exemple, chaque ligne de pixels comporte la même couleur. C'est ainsi que l'on peut coder chaque ligne de pixels par une seule paire de parenthèses.

Dans l'exemple du drapeau de la République tchèque, par contre, la couleur change à l'intérieur de chaque ligne de pixels et ceci exactement une fois : du bleu au blanc ou du bleu au rouge. Le codage pour un fichier comprimé de ce drapeau requiert donc pour chaque ligne de pixels deux paires de parenthèses. Pour le codage du fichier comprimé du drapeau français, on a besoin de trois paires de parenthèses car la couleur change dans ce cas-là deux fois à l'intérieur de chaque ligne de pixels.

Le drapeau suédois comporte une bande jaune horizontale. Pour ce secteur, on a donc besoin d'une paire de parenthèses par ligne de pixels. Dans toutes les autres lignes de pixels, la couleur change deux fois et on a besoin de trois paires de parenthèses. Il s'ensuit que le codage de l'image du drapeau suédois nécessite plus de paires de parenthèses que celui du drapeau arménien, mais moins que celui du drapeau français.

Finalement, il nous faut comparer le codage de l'image du drapeau suédois avec celui du drapeau tchèque. Si le drapeau suédois avait autant de lignes de pixels avec une couleur qu'avec trois couleurs, il faudrait en moyenne exactement 2 paires de parenthèses pour le codage. Comme la bande dans la zone moyenne du drapeau suédois est plus étroite que les autres parties du drapeau, le codage nécessite plus que deux paires de parenthèses. Il en résulte que, dans son ensemble, le codage du drapeau suédois a besoin de plus de paires de parenthèse que le codage du drapeau tchèque.

C'est de l'informatique !

La compression des données est un domaine important en informatique. Cette opération informatique consiste à réduire la taille de transmission ou de stockage des données afin de pouvoir les transmettre plus vite via le réseau. Les algorithmes de compression peuvent accélérer de manière considérable la transmission des données à l'intérieur d'un réseau. Si, par exemple, on diffusait la musique d'une webradio sans compression, on aurait besoin d'une bande passante capable de gérer un volume de données 10 fois supérieur au même volume de données compressé de façon courante aujourd'hui. C'est la raison pour laquelle on mise sur la recherche intensive des nouveaux algorithmes de compression capables de stocker les fichiers des photos, de la musique ou des vidéos de façon encore plus efficace. Le procédé de compression présenté dans la tâche appartient au type «encodage à double longueur de plage de données numériques» (RLE = Run-length encoding). Pour plus d'informations sur ce procédé, consulter : https://www.youtube.com/watch?v=ydpNscvym_E.

Sites web et mots clés

codage, compression, algorithme de compression, fichier graphique bitmap

- https://www.youtube.com/watch?v=ypdNscvym_E
- https://fr.wikipedia.org/wiki/Run-length_encoding





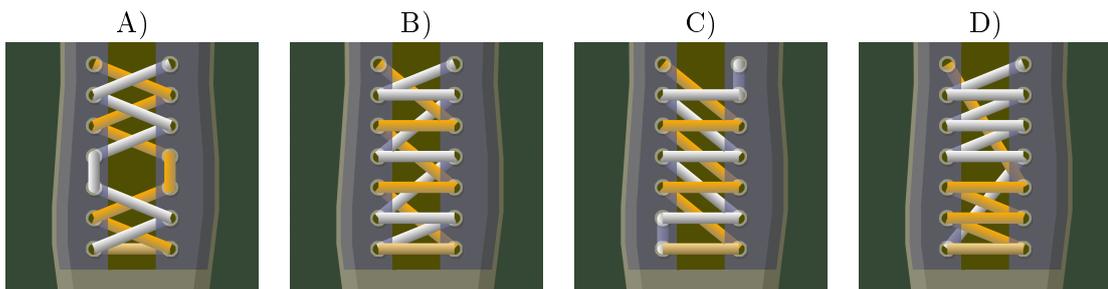
37. Las de nouer !

Les castors adorent les lacets chics. Mais comme ils sont las de les nouer, ils programment une machine qui les noue pour eux. Cette machine exécute les commandes suivantes :

Befehl	Bedeutung
{ ... }	Le contenu entre accolades est répété le plus souvent possible.
$n\{ \dots \}$	Le contenu entre accolades est répété une $n^{\text{ième}}$ fois.
orange:	Les commandes suivantes sont valides pour les lacets oranges.
blanc:	Les commandes suivantes sont valides pour les lacets blancs.
devant:	Le lacet passe devant l'œillet.
derrière:	Le lacet passe derrière l'œillet.
vers-le-haut:	Le lacet est mené vers le haut (vers le prochain œillet).
vers-le-bas:	Le lacet est mené vers le bas (vers le prochain œillet).
changement:	Le lacet est mené de gauche à droite ou l'inverse ; le lacet est ramené de manière correspondante.

Quel laçage les castors obtiennent-ils avec le programme suivant ?

```
orange: devant
blanc: devant
2{
    orange: vers-le-haut changement devant
    blanc: vers-le-haut changement devant
}
orange: vers-le-haut derrière
blanc: vers-le-haut derrière
{
    orange: vers-le-haut changement devant
    blanc: vers-le-haut changement devant
}
```





Solution

La solution correcte est la solution A). Le programme effectue le laçage des deux premiers œillets comme dans l'exemple. Ensuite, les lacets sont menés directement en haut. Finalement, les lacets changent de côté et le programme continue comme au départ.

C'est de l'informatique !

Le programme de la machine qui noue des lacets a recours à un langage de programmation simple. Ce dernier comprend des éléments caractéristiques du langage de programmation :

- Une série d'instructions effectuées les unes à la suite des autres (une *séquence*).
- Une répétition des blocs d'instructions (une *boucle*) d'un certain nombre de fois, et ceci, aussi bien en tant que boucle de comptage ainsi qu'en tant que boucle comportant des conditions d'arrêt.

À ces éléments caractéristiques du langage de programmation mentionnés ci-dessus s'ajoutent souvent deux autres éléments : l'exécution des blocs de base sous conditions particulières (*sélection*) et la sauvegarde des blocs de base fréquemment utilisés en tant que blocs séparés (*la programmation modulaire*), ce qui, par contre, n'est pas le cas dans notre exemple.

D'ailleurs, la machine qui noue des lacets peut être également programmée à travers un simulateur déjà existant. Pour ce faire, consultez le site : <http://www.klbg.com/lacing/>.

Sites web et mots clés

boucle, instructions, simulateur, lacets

- [https://fr.wikipedia.org/wiki/Interprète_\(informatique\)](https://fr.wikipedia.org/wiki/Interpr%C3%AAte_(informatique))
- https://fr.wikipedia.org/wiki/Langage_de_programmation
- <http://www.klbg.com/lacing/>
- <http://www.fieggen.com/shoelace/lacingmethods.htm>



38. Peinture récursive

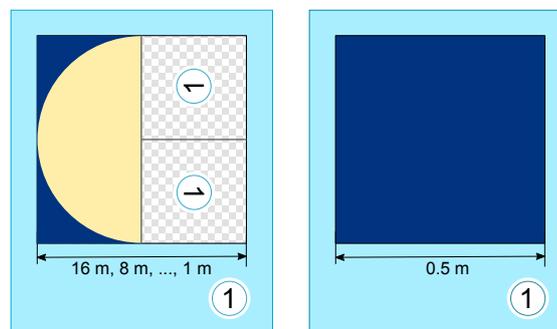
Tina et Ben sont en train de préparer une exposition temporaire au Musée de l'informatique. L'objectif est d'effectuer une peinture de 16 mètres sur 16 mètres sur le sol de l'espace de l'exposition temporaire.

De la part de l'artiste, Tina et Ben ont reçu un set de cartes. Ces cartes comportent des instructions formulées dans le langage pictural propre à l'artiste ainsi que des indications concernant les éléments de l'image, les mesures et les rotations nécessaires à effectuer. Quelques cartes comportent en outre des cases numérotées qui renvoient aux autres cartes du set.

Voici un exemple tiré d'un projet de peinture précédent. L'assemblage correct de tous les éléments montre l'image d'un castor. À noter en particulier : les longueurs différentes des cartes.

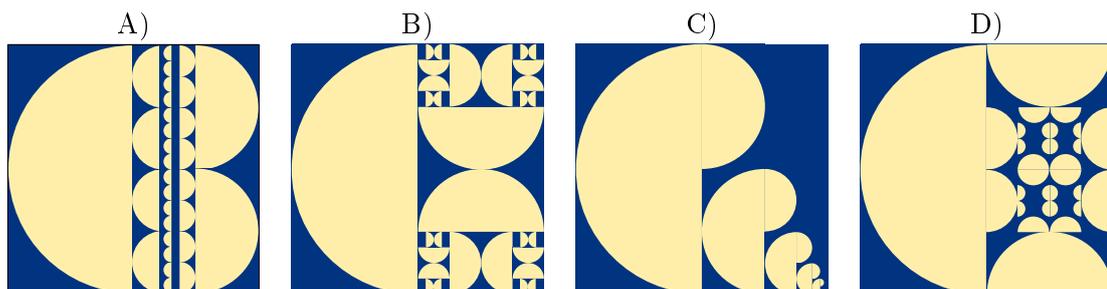


Afin de réaliser la peinture de l'exposition temporaire, Tina et Ben reçoivent deux cartes comportant les instructions suivantes :



Ben s'étonne : «Je ne vois pas comment réaliser ce projet. La carte de gauche se réfère à elle-même et puis, sur les deux cartes figure exactement le même numéro!». Tina éclate de rire : «On va y arriver, tu verras ! D'abord, nous ne nous servons que de la carte de gauche. La carte de droite nous indiquera ensuite le moment où nous devons achever l'œuvre».

Laquelle des quatre œuvres ci-dessous figurera sur le sol de l'espace de l'exposition ?





Solution

La carte de gauche nous indique qu'il faut peindre un demi-cercle sur la moitié gauche du sol. Le côté rond du demi-cercle est tourné vers la gauche. Pour peindre la moitié droite du sol, on se servira de la même carte deux fois, mais cette fois-ci, en format réduit. L'orientation des éléments de l'image correspondra à l'orientation des cases numérotées 1.

Sur la carte de gauche, la case numérotée 1 d'en haut montre le chiffre 1 tourné de 90 degrés vers la gauche. C'est la raison pour laquelle l'élément de l'image doit également être tourné de 90 degrés vers la gauche. Il s'ajoute que la rondeur du demi-cercle sera tournée vers le bas.

Sur la carte de gauche, la case numérotée 1 d'en bas montre le chiffre 1 tourné de 90 degrés vers la droite. C'est la raison pour laquelle l'élément de l'image doit également être tourné de 90 degrés vers la droite. La rondeur du demi-cercle sera tournée, cette fois-ci, vers le haut.

Toutes ces prémisses ne correspondent qu'à l'image B). C'est pourquoi B) est l'unique réponse possible.

C'est de l'informatique !

Dans le domaine de l'informatique, on nomme un programme informatique «récursif» quand les instructions s'appellent mutuellement afin que le programme puisse faire référence à lui-même et demander sa propre exécution au cours de son déroulement. Le mot «récursif» trouve son origine dans le terme latin *recurrere* (ce qui signifie «revenir en arrière», ici : «faire référence à soi-même»). Le concept récursif est très puissant. Il est possible d'accomplir certaines tâches complexes à l'aide d'un tel programme, et ceci de manière très efficace et intelligible. Pourtant, certains programmes récursifs ne sont pas aussi évidents et plutôt difficiles à comprendre.

Une instruction à fonction récursive comporte non seulement l'appel à la fonction récursive mais aussi une condition qui détermine le moment de l'interruption de cette fonction. Sinon il se peut que le programme récursif répète les instructions à l'infini (ou jusqu'à l'épuisement d'une des ressources comme de la mémoire vive (RAM)).

Sites web et mots clés

programmer, programme récursif, conditions d'arrêt

— <https://fr.wikipedia.org/wiki/Récurtivité>



39. Le jeu de formes géométriques

Alicia joue à un jeu de formes géométriques. À chaque nouvelle partie de jeu, elle remplace les formes selon de nouvelles règles qu'elle s'est imposées. Elle commence toujours avec une seule forme. Voici un exemple d'un ensemble des règles de remplacement qu'elle va appliquer :



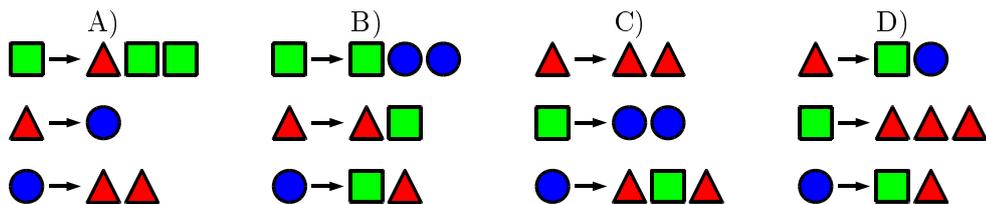
Si Alicia commence avec un carré et qu'elle respecte les deux règles ci-dessus, elle devrait procéder comme suit :



Dans la partie de jeu suivante, elle arrange les formes de cette manière :



Choisis l'ensemble des règles de remplacement qu'elle a appliqué pour arriver à l'arrangement des formes ci-dessus.





Solution

La réponse B) est correcte. Si Alicia commence avec un triangle, elle effectuera les remplacements suivants :



Les autres réponses ne sont pas correctes. Voici pourquoi :

Prenons la réponse A) : si Alicia commence avec un triangle ou un cercle, elle n'arrivera jamais à les remplacer par des carrés. Si, par contre, elle commence avec un carré, elle produira l'arrangement suivant :



Comme tous les remplacements qui s'ensuivent deviennent de plus en plus longs, cet ensemble de règles de remplacement n'aboutira pas au résultat souhaité.

Si elle applique l'ensemble des règles de remplacement C) et qu'elle commence avec un triangle, elle ne peut pas le remplacer par un carré ou un cercle. Si elle commence avec un carré, elle obtient l'arrangement suivant :



Malheureusement, elle n'arrivera jamais à remplacer les deux premiers triangles initiaux par un triangle et un carré, ce qui, pourtant, sera exigé. Voyons à quel arrangement elle arrive, si elle commence avec un cercle :



Alicia rencontrera le même problème irrésolu. Elle ne pourra jamais remplacer les deux triangles initiaux par un triangle et un carré.

Si elle applique l'ensemble des règles de remplacement D), Alicia n'arrivera pas à poser deux cercles l'un à côté de l'autre. Il s'ajoute qu'il ne sera même pas possible de reproduire plus qu'un seul cercle car chaque cercle reproduit sera tout de suite remplacé par un carré et un triangle. Cependant, pour arriver à l'arrangement souhaité, Alicia aurait besoin de deux cercles posés l'un à côté de l'autre.

C'est de l'informatique !

Les règles de ce jeu représentent dans le domaine de l'informatique théorique un système de règles avec le but de transformer des mots. Plus concrètement, on pourrait dire qu'il s'agit d'un système grammatical indépendant du contexte dans lequel on remplace un mot (dans notre exemple : un symbole géométrique) par un autre ou plusieurs mots (symboles géométriques).

Le système grammatical indépendant du contexte se compose :

- d'un ensemble de mots (dans notre exemple, il s'agit de symboles ,  et  ; en outre, on fait la distinction entre les mots que l'on remplacera plus tard et ceux avec lesquels on n'a pas le droit de les remplacer.
- d'un symbole initial (qui peut être choisi indépendamment de l'ensemble de mots).
- d'un ensemble de productions (dans notre exemple, il s'agit de l'ensemble des règles de remplacement).



Dans le système de la grammaire indépendante du contexte et contrairement à la démarche d'Alicia, on n'est pas contraint d'appliquer tous les ensembles de productions possibles en même temps. Généralement, on choisit le moment où l'on veut appliquer une des productions.

On met en pratique le système de la grammaire indépendante dans de différents domaines : pour des langages formels comme le langage de programmation ou pour la description des langages naturels ou, par exemple, pour décrire le processus de croissance des plantes.

Pour résoudre le problème, il faut donc, d'une part, comprendre les règles imposées et d'autre part, savoir analyser des chaînes de symboles. Dans le domaine de l'informatique, ce concept est nommé «parsing», c'est-à-dire un concept de l'analyse syntaxique. Le *parsing* s'applique pour traduire un programme d'ordinateur, écrit dans un langage de programmation, de manière que l'ordinateur puisse l'exécuter.

Sites web et mots clés

grammaire indépendante du contexte, analyse syntaxique

- https://fr.wikipedia.org/wiki/Grammaire_non_contextuelle
- <https://fr.wikipedia.org/wiki/L-Systeme>
- https://fr.wikipedia.org/wiki/Analyse_syntaxique





40. Les écureuils égoïstes

Les écureuils vivent dans les cavités des arbres. Un arbre a cinq cavités disposées les unes au-dessus des autres. Ces cavités sont peuplées de seize écureuils ce qui revient à dire que ces seize écureuils habitent ensemble les cinq cavités d'arbre.

Chaque matin, tous les écureuils vérifient les cavités pour savoir dans lesquelles se trouvait le nombre le plus restreint des écureuils la nuit précédente. Plus exactement, les écureuils contrôlent le nombre de voisins qui demeuraient au-dessus ou au-dessous de leur propre cavité. Pour la nuit suivante, chaque écureuil choisit en



clandestin la cavité avoisinée où il y avait le plus petit nombre d'écureuils pendant la nuit précédente. Au cas où chaque cavité hébergeait le même nombre d'écureuils, les écureuils préfèrent passer la nuit plutôt dans leur propre cavité que dans la cavité située au-dessus de cette-dernière, et s'ils ne passent pas la nuit dans leur propre cavité, ils préfèrent la cavité située au-dessus à celle située au-dessous de leur propre cavité.

Si, par exemple, pendant la nuit précédente se trouvaient dans les cavités de haut en bas 5, 0, 0, 4 et 7 écureuils, le jour suivant, la situation changera comme suit : Tous les cinq écureuils qui ont passé la nuit dans la cavité située tout en haut se déplaceront pour passer la nuit dans la cavité située directement au-dessous de leur propre cavité (car ils préfèrent avoir 0 à 4 voisins). Les 7 écureuils qui ont passé la nuit tout en bas de la séquence des cavités se déplaceront pour passer la nuit dans la cavité située directement au-dessus de leur propre cavité (car ils préfèrent avoir 4 à 6 voisins) et les 4 écureuils qui ont passé la nuit dans la cavité au-dessus de celle des 7 écureuils se déplaceront pour passer la nuit dans la cavité située directement au-dessus de leur propre cavité (car ils préfèrent avoir 0 à 3 voisins).

Ce matin, les cavités étaient peuplées comme suit : de haut en bas, il y avait 6, 3, 3, 0 et 4 écureuils. Combien de jours faudra-t-il pour que tous les écureuils se retrouvent de nouveau exactement dans les mêmes cavités qu'aujourd'hui ?

- A) Deux jours.
- B) Trois jours.
- C) Quatre jours.
- D) Pas de chance, ils ne se retrouveront plus jamais dans la même cavité.



Solution

La réponse correcte est : trois jours.

Aujourd'hui : 6, 3, 3, 0, 4
Le jour suivant : 0, 9, 0, 7, 0
Après 2 jours : 9, 0, 7, 0, 0
Après 3 jours : 0, 16, 0, 0, 0

C'est de l'informatique !

Le problème en question est un exemple de l'intelligence collective. L'idée de base de ces algorithmes est que l'on puisse résoudre des problèmes complexes avec des éléments très simples à condition que l'on en dispose de beaucoup. Les fourmis, par exemple, se comportent selon des règles simples et indépendamment les unes des autres. Cependant, quand il y en a beaucoup, elles sont capables d'accomplir des tâches exigeantes comme par exemple construire une fourmilière et y aménager des dômes, trouver le chemin le plus favorable afin d'atteindre une ressource ou couper des feuilles.

Dans la tâche présente, il y a également une grande «quantité» d'éléments (dans notre cas, elle est représentée par des écureuils) qui se comportent selon des règles simples. Malheureusement, leur comportement collectif est loin d'être perçu comme «intelligent». Ils revendiquent tous l'espace le plus grand possible, pourtant, à la fin, ils se retrouvent tous dans la même cavité. Il s'ensuit donc que l'on ne peut pas attribuer le comportement des fourmis à un algorithme de colonies de fourmis à 100% : parfois on s'aperçoit toutefois qu'il est mieux de coopérer au lieu de se comporter de manière égoïste.

Sites web et mots clés

intelligence collective, algorithme de colonies des fourmis

- https://fr.wikipedia.org/wiki/Intelligence_distribuée
- https://fr.wikipedia.org/wiki/Intelligence_collective
- https://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis



A. Auteurs des exercices

 Nursultan Akhmetov	 Hans-Werner Hein	 Erkulan Nurtazanov
 Rosa Alexos	 Mathias Hiron	 Henry Ong
 Guðjón Karl Arnarson	 Sarah Hobson	 Serena Pedrocchi
 Wilfried Baumann	 Martin Horvath	 Wolfgang Pohl
 Daphne Blokhuis	 Juraj Hromkovič	 Ilya Posov
 Ivo Blöchliger	 Yukio Idosaka	 Sergei Pozdniakov
 Andrea Brabcová	 Mile Jovanov	 Dániel Pressing
 Eugenio Bravo	 Martina Kabátová	 J. P. Pretti
 Nicolas Brunner	 Filiz Kalelioğlu	 Lorenzo Repetto
 Alexander Cirri	 Joseph Kaperst	 Kirsten Schlüter
 Valentina Dagienė	 Akiko Kikui	 Eljakim Schrijvers
 Christian Datzko	 Ries Kock	 Sue Sentance
 Susanne Datzko	 Ágnes Kocsis	 Maiko Shimabuku
 Janez Demšar	 Tobias Kohn	 Emil Stankov
 Marissa Engels	 Ivana Kosírová	 Björn Steffen
 Olivier Ens	 Bernd Kurzmann	 Gabrielė Stupurienė
 Hanspeter Erni	 Greg Lee	 Seiichi Tani
 Jürgen Frühwirth	 Dan Lessner	 Peter Tomcsányi
 Gerald Futschek	 Hiroki Manabe	 Monika Tomcsányiová
 Peter Garscha	 Khairul A. Mohamad Zaki	 Willem van der Vegt
 Haris Gavranovic	 Hamed Mohebbi	 Jiří Vaníček
 Yasemin Gülbahar	 Nataša Mori	 Troy Vasiga
 Martin Guggisberg	 Anna Morpurgo	 Lina Vinikienė
 Urs Hauser	 Tom Naughton	 Michael Weigend



B. Sponsoring : Concours 2016

HASLERSTIFTUNG <http://www.haslerstiftung.ch/>



<http://www.roborobo.ch/>



<http://www.digitec.ch/>



<http://www.baerli-biber.ch/>



<http://www.verkehrshaus.ch/>
Musée des transports, Lucerne



**Kanton Zürich
Volkswirtschaftsdirektion
Amt für Wirtschaft und Arbeit**

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich



i-factory (Musée des transports, Lucerne)



<http://www.ubs.com/>



<http://www.bbv.ch/>



<http://www.presentex.ch/>



ITgirls@hslu

<https://www.hslu.ch/de-ch/informatik/agenda/veranstaltungen/fuer-schulen/itgirls/>
HLSU, Lucerne University of Applied Sciences and Arts
Engineering & Architecture

PH LUZERN
PÄDAGOGISCHE
HOCHSCHULE

<http://www.phlu.ch/>
Pädagogische Hochschule Luzern

ABZ

AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

<http://www.abz.inf.ethz.ch/>
Ausbildungs- und Beratungszentrum für Informatikunterricht der
ETH Zürich.



C. Offres ultérieures

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuisse del'inform
atique dans l'enseignement//societàsviz
zeraperl'informaticanell'insegnamento

Devenez vous aussi membre de la SSIE
<http://svia-ssie-ssii.ch/la-societe/devenir-membre/>

et soutenez le Castor Informatique par votre adhésion

Peuvent devenir membre ordinaire de la SSIE toutes les personnes qui enseignent dans une école primaire, secondaire, professionnelle, un lycée, une haute école ou donnent des cours de formation ou de formation continue.

Les écoles, les associations et autres organisations peuvent être admises en tant que membre collectif.