



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

Exercices et solutions 2017 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
schweizerischerverein für informatik in d
er ausbildung // société suisse de l'inform
atique dans l'enseignement // società sviz
zera per l'informatica nell'insegnamento



Ont collaboré au Castor Informatique 2017

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

Nous adressons nos remerciements à :

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Regula Lacher, Ivana Kosírová : ETHZ

Valentina Dagiene : Bebras.org

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

Anna Morpurgo, Violetta Lonati, Mattia Monga : Italie

Gerald Futschek, Wilfried Baumann : Austrian Computer Society, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Eljakim Schrijvers, Daphne Blokhuis : Eljakim Information Technology bv, 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 2017 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 2017. Ce cahier d'exercice a été produit le 18 novembre 2017 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. 108.



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 pour 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 scolaires 3 et 4) a été organisé pour la première fois en 2012.

Le Castor Informatique vise à motiver les élèves à apprendre 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 quant à l'utilisation des ordinateurs, sauf de savoir naviguer sur Internet, car le concours s'effectue en ligne. 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 2017 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 (5 faciles, 5 moyens et 5 difficiles).

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 de son 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 2017	i
Préambule	ii
1. Des places de parking libres	1
2. Nichoirs pour oiseaux	3
3. Trouve le passage !	5
4. Le portail binaire	7
5. Service de transmission de messages	9
6. La chasse à la fraise	11
7. Déplacer des chiens	13
8. Le castor à un bras	15
9. Enlever des murs	19
10. Cinq petits bouts de bois	23
11. Concours de danse : à qui le tour ?	27
12. Le nom japonais	31
13. Un programme court	33
14. Orner des pendentifs médiévaux	35
15. Le journal scolaire	37
16. Honomakato	41
17. Un art martial japonais	45
18. La confiture de grand-père	47
19. La ville riche en ronds-points	49
20. Pizzeria Castoria	51
21. Une commande chiffrée	55
22. Jeu des pièces	57
23. Bar à jus de fruits	59
24. Intrusion au musée	63

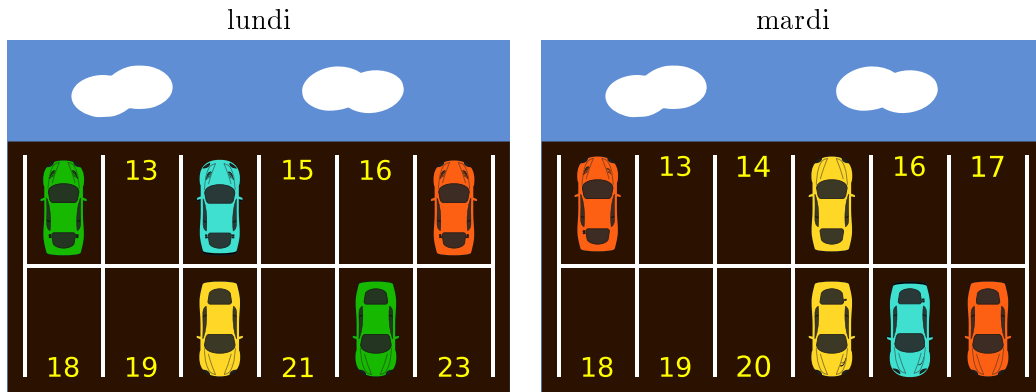


25. Des jeux de lumière	65
26. Substitutions	67
27. Sors du labyrinthe !	69
28. Système d'arrosage	71
29. Une nouvelle chanson	73
30. Jeu de billes	75
31. À table, mais vite !	79
32. Aide l'Arabot !	83
33. Les piles de cure-dents à diviser	85
34. Calculer la distance entre les mots	87
35. Des téléchargements en parallèle	89
36. Accumuler des points	93
37. La méthode Quadtree	97
38. Raccourci ou détour ?	101
39. L'affichage numérique	103
40. Subdivision du code	105
A. Auteurs des exercices	108
B. Sponsoring : Concours 2017	109
C. Offres ultérieures	111



1. Des places de parking libres

Le parking des castors permet de stationner 12 voitures. Chaque place de parking est numérotée. Les images ci-dessous montrent les places de parking qui étaient occupées lundi dernier et celles qui étaient occupées mardi dernier.



Combien de places de parking étaient libres ces deux journées, lundi et mardi ?

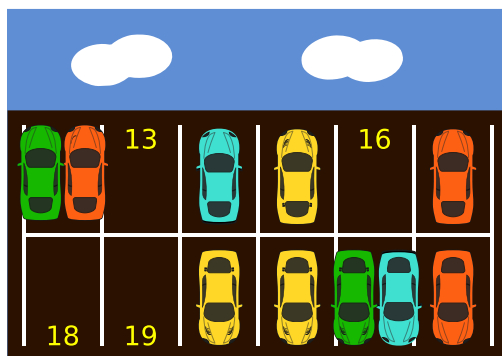
- A) 3
- B) 4
- C) 5
- D) 6



Solution

La réponse B) est correcte puisque les places 13, 16, 18 et 19 sont restées libres pendant les deux journées en question.

Quand on représente les voitures garées sur un seul parking, nous voyons tout de suite quelles places sont restées libres. En les additionnant, on arrive à quatre places qui n'ont pas été occupées, ni lundi ni mardi.



C'est de l'informatique !

Il est possible d'encoder tous les types de données sous forme d'une série de «zéros» et de «uns». Une telle série de données s'appelle *code binaire*. Chaque zéro et chaque un est dénommé *bit* («binary digit»).

Dans la tâche présente, nous pouvons encoder la présence d'une voiture sur le parking sous forme de un (1) et l'absence d'une voiture sous forme de zéro. L'occupation du parking peut ainsi être représentée sous forme d'une série de *bits* : celle de lundi correspondra donc à la série 101001001010 et celle de mardi à la série 100100000111. À l'aide d'un opérateur logique *OU (OR)* qui, lui, indique que le résultat sera VRAI si la valeur A OU la valeur B sont vraies, toutes les places qui sont occupées au moins pendant une des deux journées seront encodées sous forme de un (1). Par la suite, nous calculons la réponse correcte en superposant les deux séries de données et nous obtenons le résultat suivant :

$$\begin{array}{r}
 101001001010 \\
 \text{ou} \\
 100100000111 \\
 = \\
 101101001111
 \end{array}$$

Ce code binaire comporte quatre zéros ce qui correspond à quatre places de parking libres.

Sites web et mots clés

bits, binaire, code binaire, OU (OR), opérateur logique

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



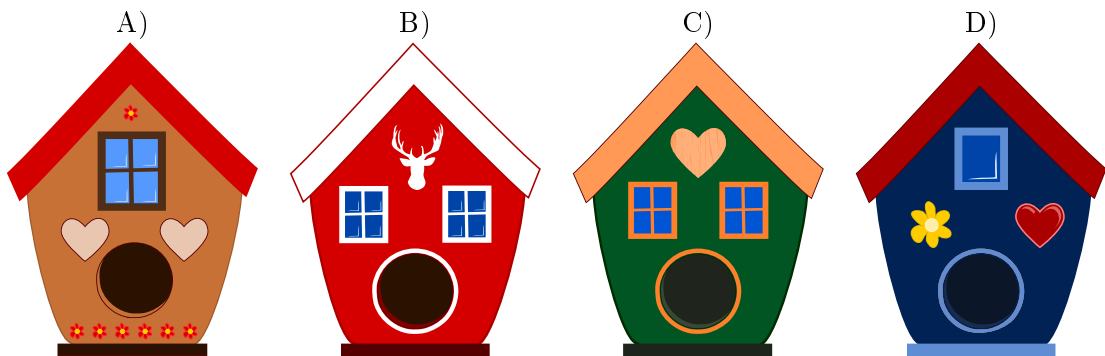
2. Nichoirs pour oiseaux

Maman castor aimerait bien acheter un nichoir pour sa fille qui fêtera son anniversaire demain. Pour ne pas se tromper de nichoir, elle demande à sa fille lequel des nichoirs lui ferait plaisir. Sa fille lui répond :

«J'aimerais un nichoir avec deux fenêtres et un coeur.»

La mère va donc au magasin pour animaux acheter un nichoir.

Lequel de ces quatre nichoirs maman castor achètera-t-elle pour sa fille ?





Solution

La réponse C) est correcte car le nichoir comporte 2 fenêtres et 1 coeur.

La réponse A) est fausse parce que le nichoir n'a qu'une seule fenêtre. En outre, il a 2 coeurs au lieu d'un seul.

La réponse B) est fausse parce que le nichoir comporte 2 fenêtres, mais aucun coeur.

La réponse D) est fausse parce que le nichoir ne comporte qu'une seule fenêtre.

C'est de l'informatique !

Cette tâche est difficile car il faut se concentrer sur un nichoir qui corresponde aux désirs de la fille de maman castor. Pour choisir le bon nichoir, il faut donc ignorer toutes les autres caractéristiques des nichoirs comme par exemple la couleur de la maison ou d'autres détails qui ne sont pas importants. Cette vision «limitée» est appelée abstraction, ce qui veut dire que l'on ne prendra en considération que des propriétés préalablement déterminées. En informatique, ceci est souvent nécessaire.

Sites web et mots clés

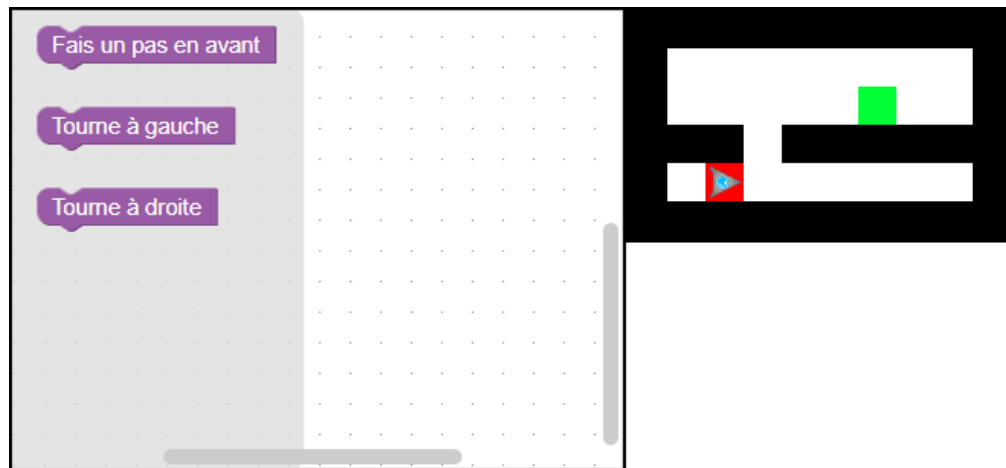
modèle (pattern), reconnaissance des formes (pattern recognition), propriété, abstraction (informatique)

- <https://en.wikipedia.org/wiki/Abstraction>
- https://en.wikipedia.org/wiki/Pattern_recognition



3. Trouve le passage !

Un robot ayant une forme triangulaire se trouve sur la position de départ marquée en rouge. Il doit atteindre la zone cible marquée en vert.

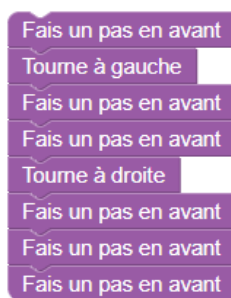


Pour programmer le robot afin qu'il arrive jusqu'à la zone cible, tu dois insérer des instructions dans la fenêtre de programme tout en respectant l'ordre correct de la séquence d'instructions.



Solution

Voici une solution possible :



Pour que le robot atteigne la zone cible, il doit d'abord avancer d'un pas afin de se trouver à côté de la brèche dans le mur. A partir de là, s'il tourne à gauche et avance de deux pas, il traverse la brèche. Finalement, s'il tourne à droite et avance de trois pas, il arrive directement à la zone cible.

C'est de l'informatique !

En robotique «mobile», la navigation pose toujours certains problèmes. La résolution de problèmes tels que celui de cet exercice consistant à devoir sortir d'un labyrinthe n'est pas si courante, elle exige des facultés de pensée computationnelle. Pour résoudre ce problème, on se sert d'un robot automatisé. Il existe différents types de labyrinthes : avec ou sans boucle, avec ou sans quadrillage. Le robot dans notre tâche ne possède pas de capteurs qui lui permettraient de détecter les murs. Il est quasiment aveugle. C'est la raison pour laquelle il dépend d'instructions très précises pour trouver le passage et, par conséquent, le bon chemin pour arriver à la zone cible.

Sites web et mots clés

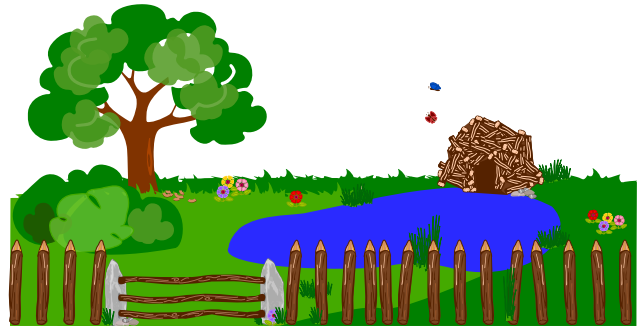
séquence, programmer

- <https://fr.wikipedia.org/wiki/Blockly>
- <http://primalogo.ch/fr>
- <http://www.abz.inf.ethz.ch/primarschulen-stufe-sek-1/programmieren-fur-kinder/>
- http://www.swisseduc.ch/informatik/programmiersprachen/scratch_werkstatt/
- <http://ilearnit.ch/de/2b/explain.html>
- <https://scratch.mit.edu/>
- <http://kinderlabor.ch/informatik-fuer-kinder/programmieren-mit-scratch/>



4. Le portail binaire

Les castors ont l'habitude de se rendre visite assez régulièrement. Cependant, il arrive que l'un ou l'autre ne soit pas chez lui et que les castors arrivent en vain. Afin que le castor qui a quitté sa maison puisse laisser un message, par exemple pour informer ses amis quand il reviendra, les castors ont inventé une méthode très pratique. Le portail du jardin comporte deux piliers en pierre et trois bâtons en bois que l'on peut placer dans des paires de trous prédéterminés qui se trouvent dans les faces opposées des deux piliers en pierre. Cette construction permet de créer des messages courts.



Les castors se sont mis d'accord sur quatre messages :



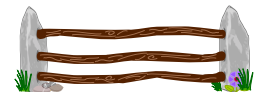
Nous sommes à la maison, venez nous voir.



Nous serons de retour à midi déjà.



Nous serons de retour dans la soirée.



Nous passons une soirée avec nos amis et nous serons de retour vers minuit.

Il serait parfaitement possible que les castors se mettent d'accord sur d'autres messages encore, sans qu'ils aient besoin ni d'autres bâtons en bois ni de trous supplémentaires.

Devine, avec deux piliers en pierre, trois bâtons en bois et six trous, sur combien de messages différents les castors pourraient-ils se mettre d'accord au total (bien entendu, y compris les quatre messages mentionnés ci-dessus).



Solution

Les castors peuvent se mettre d'accord sur huit messages différents au total :



C'est de l'informatique !

Dans la présente tâche, les castors se servent en fait d'un système de numération appelé binaire à trois positions. Les porteurs d'information sont représentés par les paires de trous qui se trouvent dans les faces opposées des piliers. Une telle paire de trous représente deux valeurs, à savoir «bâton en bois placé» et «bâton en bois non placé». La position des paires de trous nous renseigne donc sur sa valeur. Ainsi, le nombre des différents messages correspondra au nombre des valeurs des paires de trous (2) multiplié par le nombre des paires de trous (3), donc : $2^3 = 2 \cdot 2 \cdot 2 = 8$.

En général, les castors comprennent leurs messages amicaux et si jamais il arrive qu'ils se trompent, il n'y aura pas de graves conséquences. En informatique, par contre, étant donné qu'il existe différents systèmes globaux interconnectés, toutes les personnes impliquées dépendent d'une communication simple et correcte qui exclut les malentendus.

De grandes organisations s'occupent de la standardisation des systèmes de numération et de la certification des produits. Des comités d'experts multinationaux discutent et décident du caractère et du sens des signes, puis, les parlements de nombreux états approuvent ces décisions et intègrent les systèmes de numération validés dans leur législation. Voilà comment beaucoup d'hommes prennent soin que tous les ordinateurs du monde se comprennent.

Sites web et mots clés

codage, code binaire, standard

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

5. Service de transmission de messages

Violette aimerait bien envoyer un message à Léo, mais elle a besoin d'aide de la part des castors. Elle découpe le message original en quelques petits billets dont chacun comportera trois lettres. Chaque castor messenger recevra un de ces petits billets.

Sachant que les castors peuvent être facilement détournés de leur tâche lors du trajet entre Violette et Léo et que leur arrivée peut donc être désordonnée, Violette numérote chaque petit billet avant de le remettre à un des castors. Plus tard, après avoir reçu les billets, Léo n'aura qu'à les remettre dans le bon ordre pour lire le message complet.

Voilà un exemple : afin d'envoyer le message FETONSCESOIR, elle découpe quatre petits billets qui comportent les lettres suivantes :



L'autre jour, Léo a reçu la séquence de petits billets suivante :



Quel était le message original ?

- A) APPELELONBALORT
- B) LONBALELEORTAPP
- C) APPORTELEBALLON
- D) ELEAPPORTBALLON



Solution

La réponse correcte est C) APPORTELEBALLON.

Quand on classe les petits billets dans l'ordre correct, on reçoit le message original suivant :



C'est de l'informatique !

Afin de transmettre des données sur Internet comme par exemple des e-mails, des images ou des vidéos, celles-ci sont découpées en plusieurs entités appelées paquets de données (TCP/IP en anglais, Transmission Control Protocol/Internet Protocol). Chaque paquet ne dépasse pas la taille maximale de 65536 bits (soit $2^{16} = 65536$ bits = 64 KB).

Puis, à l'aide de différents routeurs, ces paquets numériques, souvent enrichis d'informations supplémentaires qui dévoilent les spécificités du paquet (expéditeur, destinataire, numéro de séquence, ...), seront dirigés à travers le réseau informatique. Toutes ces informations supplémentaires sont nécessaires pour que le destinataire puisse reconstruire le message original même si ce dernier est transmis sous forme de paquets partiels.

Remarque : Le nouveau *protocole Internet* «Ipv6» permet de gérer des paquets de plus grande taille encore.

Sites web et mots clés

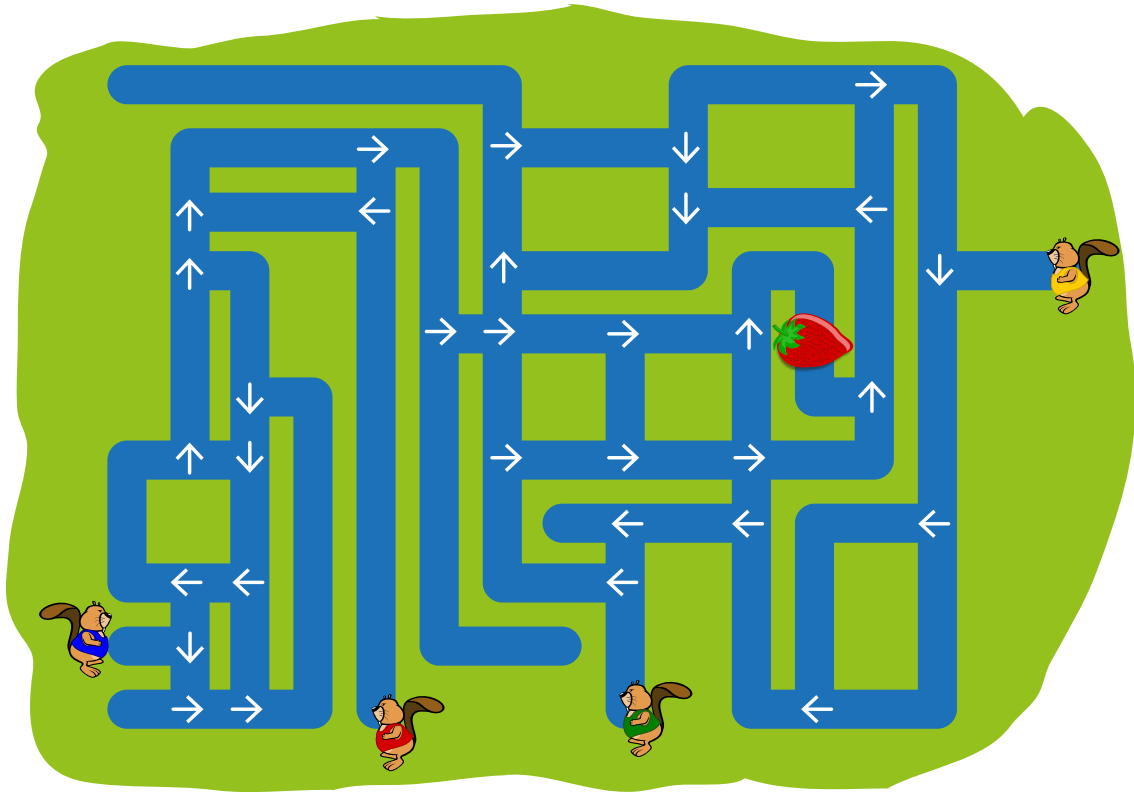
lettre, petit billet, message, numéro, ordre des petits billets, paquet de données, transmission de messages, routeur, protocole Internet

- [https://fr.wikipedia.org/wiki/Paquet_\(réseau\)](https://fr.wikipedia.org/wiki/Paquet_(réseau))
- https://en.wikipedia.org/wiki/IP_fragmentation



6. La chasse à la fraise

Quatre castors commencent à nager à partir de quatre points de départ différents. Ils nagent droit devant eux et suivent les flèches à chaque fois qu'ils arrivent à une intersection.



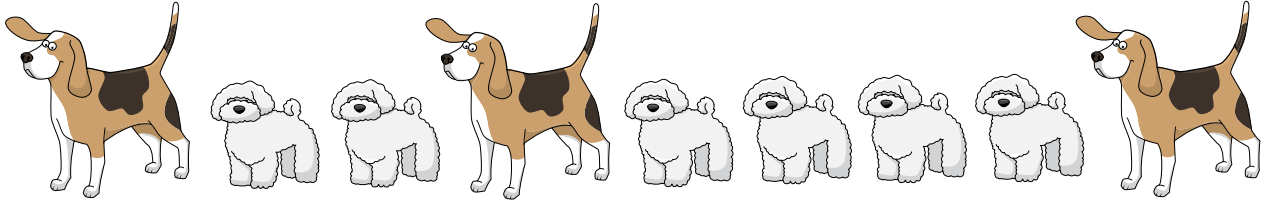
Combien de castors arriveront jusqu'à la fraise ?

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

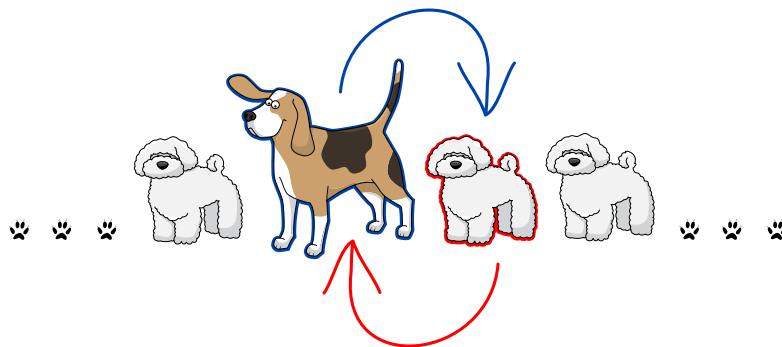


7. Déplacer des chiens

Des chiens de deux races différentes se placent en rang, l'un à côté de l'autre :



Quand deux chiens qui sont placés l'un à côté de l'autre changent leurs places, nous parlons d'un déplacement :



Suite à quelques déplacements, les trois grands chiens se retrouvent côte à côte.

Comment faut-il procéder pour déplacer les chiens le moins possible afin que les trois grands chiens se retrouvent côte à côte ?

- A) 5
- B) 6
- C) 7
- D) 8



Solution

La réponse B) (6) est correcte. D'abord, il faut déplacer le premier grand chien deux fois vers la droite. Ensuite, on déplace le dernier grand chien quatre fois vers la gauche.

Ainsi, on déplace chaque petit chien une fois puisque chaque petit chien est placé entre deux grands chiens. Le déplacement de deux petits chiens est sans effet, c'est pourquoi on préfère échanger les positions d'un petit chien avec celle d'un grand chien. Et comme il y a six petits chiens en tout, il y aura forcément au moins six déplacements.

Placer tous les grands chiens vers la droite ou vers la gauche nécessitera six déplacements supplémentaires.

C'est de l'informatique !

Afin de stocker des données, l'ordinateur utilise un ensemble de ressources appelé *mémoire* (Memory). La mémoire peut être interne (désigné par le terme RAM) ou externe (comme par exemple un disque dur ou une clé USB). Un ordinateur est équipé de plusieurs types de mémoires qu'il utilise alternativement : la mémoire interne, par exemple, permet à l'ordinateur d'accéder aux données plus rapidement que la mémoire externe. Cette dernière, par contre, est beaucoup moins chère, c'est pourquoi les ordinateurs modernes utilisent plutôt les mémoires externes qu'internes. Comme l'ordinateur peut lire et écrire plus rapidement sur la mémoire interne, les scientifiques ont intérêt à rechercher des solutions pour que l'ordinateur utilise la mémoire interne plutôt que la mémoire externe.

Dans la présente tâche, nous avons dû déplacer deux chiens c'est-à-dire deux éléments. Si nous considérons ces éléments comme des unités de données à l'intérieur de la mémoire, ce déplacement représentera l'échange de position de deux unités de données. Dans le cas où ces unités sont stockées dans une mémoire externe, il convient d'effectuer aussi peu de déplacements que possible.

Nous connaissons de tels déplacements dans le domaine de différents algorithmes de tri. Là aussi, pour que le processus soit efficace, il convient de minimiser le nombre des déplacements. Un algorithme de tri est, par exemple, le *tri à bulles*. Cet algorithme compare répétitivement des éléments consécutifs d'un *tableau* (d'une structure de données) pour les inverser s'ils sont mal triés : par exemple, il déplace les plus grandes unités de données en fin de tableau.

Sites web et mots clés

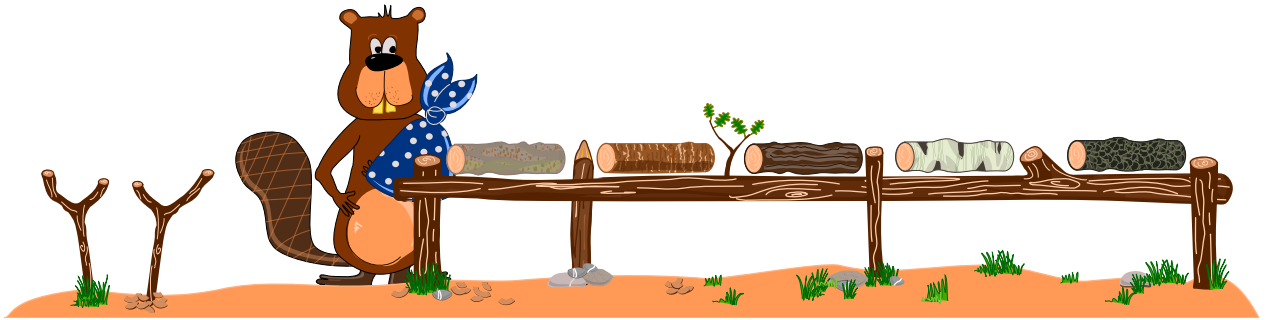
déplacement, mémoire interne, mémoire externe, tri à bulles, tableau

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



8. Le castor à un bras

Le pauvre David s'est cassé le bras gauche et ne peut travailler qu'avec le bras droit. Il aimerait bien classer sa collection de bûches de bois, mais à cause de sa blessure, il ne peut soulever qu'une bûche à la fois. Ce qu'il peut faire, pourtant, c'est déposer une bûche sur le support qui se trouve à sa gauche et qui peut lui servir d'entrepôt.

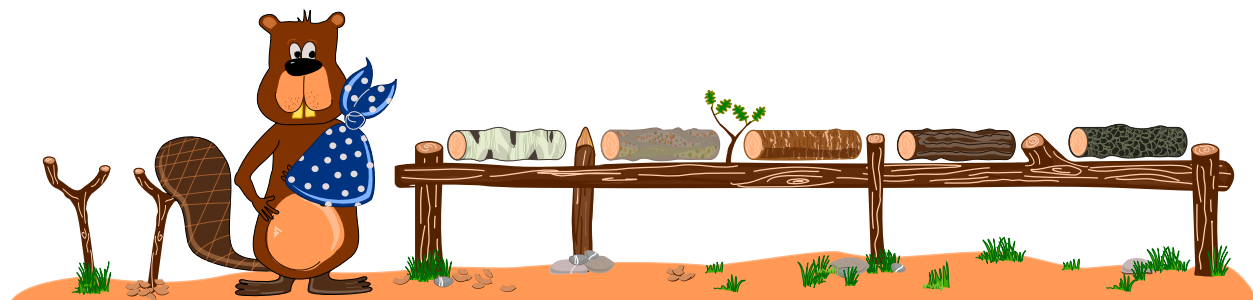


David te demande de l'aider à classer les différentes bûches selon leur couleur, de la plus claire à la plus foncée. Il aimerait bien que la bûche la plus claire se trouve à l'extrême gauche et la bûche la plus foncée à l'extrême droite.



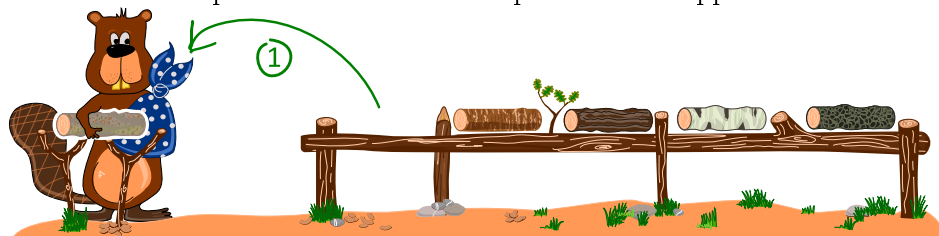
Solution

La réponse correcte est :

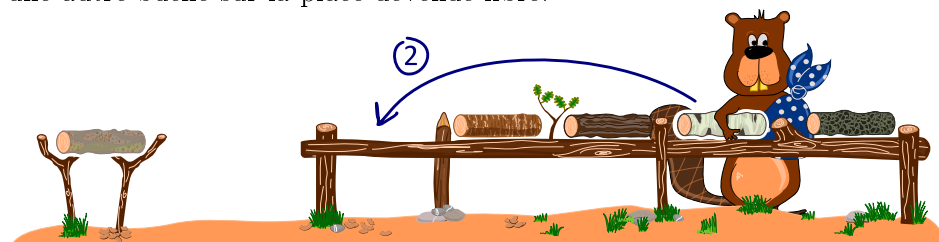


Pour que David puisse résoudre ce problème, il est nécessaire qu'il utilise le support à sa droite. Il existe plusieurs possibilités pour classer les bûches correctement, mais la plus simple est qu'il échange deux bûches en entreposant une des deux bûches sur le support :

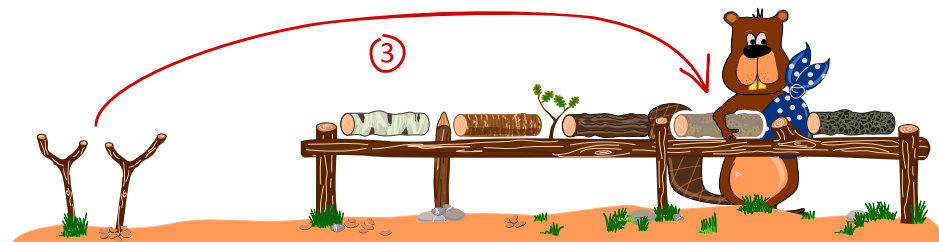
- Enlever une bûche de sa position initiale et la déposer sur le support.



- Déposer une autre bûche sur la place devenue libre.



- Reprendre la bûche qui repose actuellement sur le support et la déposer sur la place devenue libre.



Ainsi, en échangeant progressivement les bûches deux par deux, on arrivera à l'ordre désiré.

C'est de l'informatique !

En informatique, cette méthode d'échanger deux bûches de bois en utilisant une place supplémentaire comme entrepôt, est courante. Les «places» sont en général des *variables* (par exemple, appelées *a* et *b*) qui comprennent des *valeurs*. Si on veut intervertir la valeur des variables *a* et *b*, il faut faire appel à une troisième variable appelée *t*. D'abord, on enregistre temporairement la valeur de la variable *a* dans la variable *t*. Ensuite, on enregistre la valeur de la variable *b* dans la variable *a* pour enregistrer finalement la valeur de la variable temporaire *t* (qui comprend la valeur de la variable *a*) dans la variable *b*. Nous pouvons décrire cette procédure comme suit :


$$t \leftarrow a$$
$$a \leftarrow b$$
$$b \leftarrow t$$

Afin de classer un ensemble de plusieurs variables à l'aide d'une seule variable temporaire, on peut utiliser l'algorithme *tri par sélection* :

- De la première variable jusqu'à la dernière :
 - Sélectionne la plus petite valeur de l'ensemble des variables qui n'ont pas encore été triées
 - Invertis cette valeur avec la valeur de la variable actuellement en question à l'aide de la variable temporaire

Outre l'algorithme simple *tri par sélection*, il existe d'autres algorithmes de tri simples comme par exemple le *tri par insertion* ou le *tri à bulles*. Un algorithme de tri plus efficace encore mais par-delà aussi plus complexe est le tri rapide (en anglais, *quicksort*).

Sites web et mots clés

intervertir des variables, (algorithme de) tri

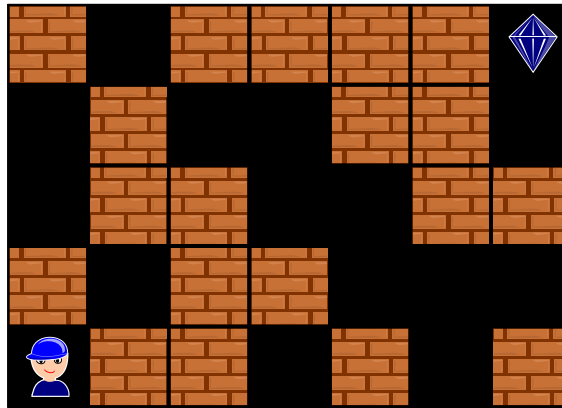
- [https://fr.wikipedia.org/wiki/Permutation_\(informatique\)](https://fr.wikipedia.org/wiki/Permutation_(informatique))
- https://fr.wikipedia.org/wiki/Algorithme_de_tri





9. Enlever des murs

Pour arriver jusqu'au trésor qui se trouve en haut à droite, Pierre doit enlever des murs. Son but est d'en enlever le moins possible.

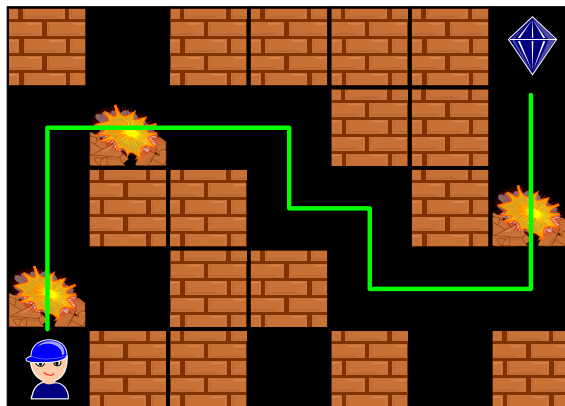


Quels murs au minimum doivent être enlevés pour libérer le chemin jusqu'au trésor ?



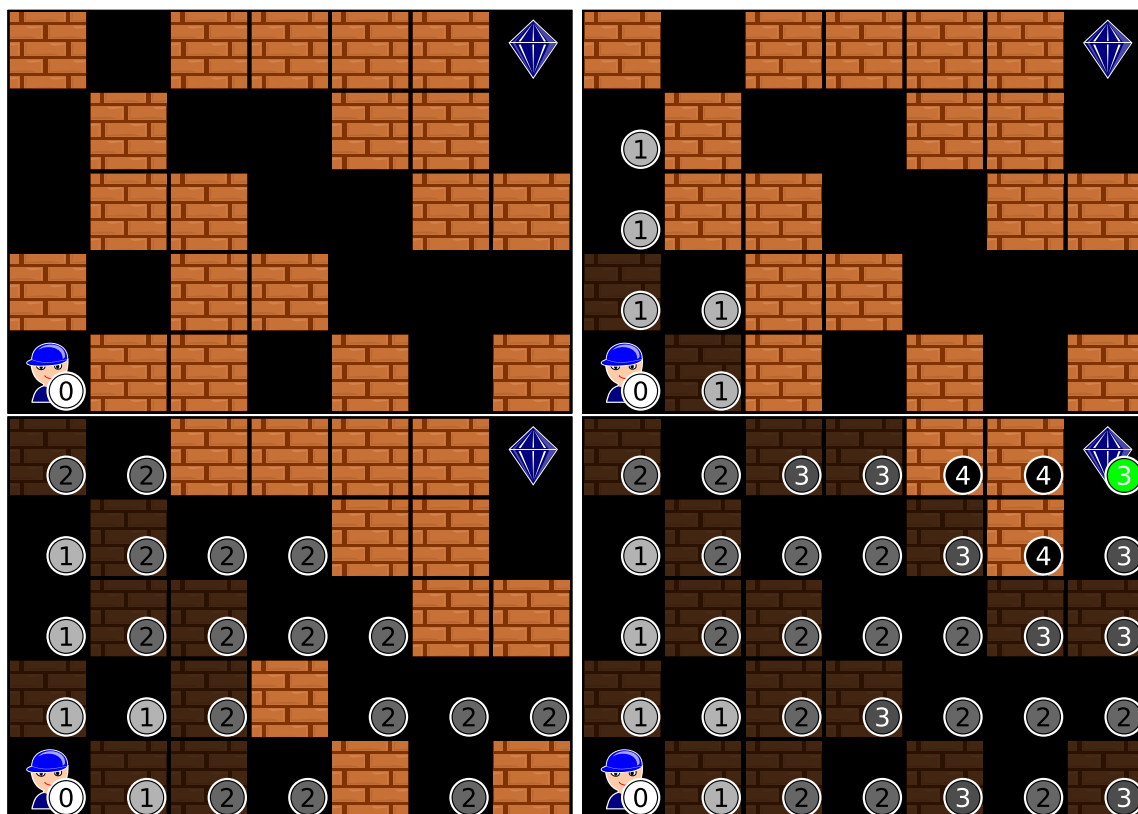
Solution

Il faut enlever au moins trois murs. La figure ci-dessous montre le nombre de murs qu'il faut enlever (les murs à enlever sont représentés par le symbole d'une petite explosion). Le chemin qui mène ainsi directement au trésor est marqué en vert.



On peut résoudre ce problème en procédant de manière systématique : on note dans chaque cellule le nombre de murs qu'il a fallu enlever pour y arriver. Nous commençons au point de départ en bas à gauche. Nous notons dans la cellule le chiffre 0. Nous marquons toutes les cellules noires que l'on peut atteindre directement à partir de ce point par le chiffre 0. Ensuite, nous marquons tous les murs avoisinant les 0 par le chiffre 1 et nous notons dans toutes les cellules noires que l'on peut atteindre en enlevant ces murs (qui avoisinent les 0) le chiffre 1.

Puis, nous marquons tous les murs et toutes les cellules que l'on peut atteindre en enlevant deux murs par le chiffre 2, et toutes les cellules que l'on peut atteindre en enlevant 3 murs par le chiffre 3, et ainsi de suite. En procédant de cette manière, nous noterons finalement dans la zone cible le chiffre 3 qui, lui, correspondra à notre solution mentionnée ci-dessus.





C'est de l'informatique !

Le problème du plus court chemin dans un labyrinthe est un problème informatique très répandu. Les systèmes de navigation, par exemple, se servent de différentes techniques informatiques pour trouver le chemin le plus court, tout en respectant des conditions particulières comme par exemple d'éviter des routes à péage. Dans notre cas, il faut simplement trouver le chemin le plus court à travers les murs, la longueur réelle des chemins n'a pas d'importance.

Pour trouver la solution à notre problème, il faut un raisonnement logique et une démarche (un algorithme) qui tient compte des résultats intermédiaires. En procédant ainsi, on fait appel à un algorithme particulier appelé *algorithme de parcours en largeur*.

Sites web et mots clés

labyrinthe, plus court chemin, algorithme de parcours en largeur.

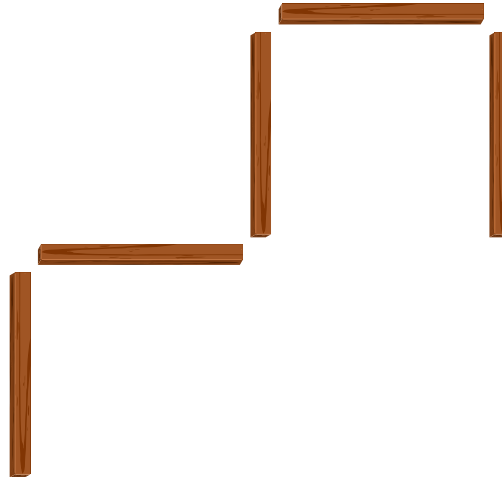
- https://fr.wikipedia.org/wiki/Problème_de_plus_court_chemin
- https://fr.wikipedia.org/wiki/Algorithme_de_parcours_en_largeur



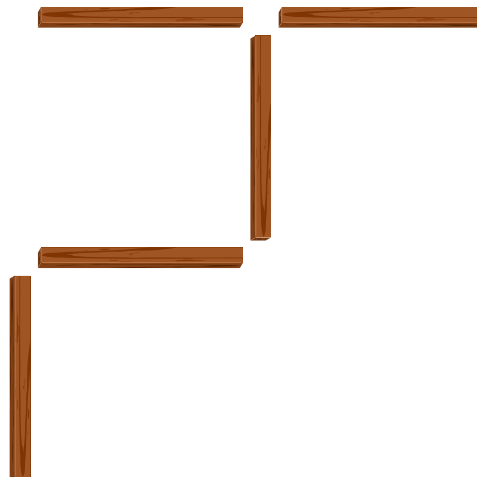


10. Cinq petits bouts de bois

Sur une table, il y a 5 petits bouts de bois arrangés de la manière suivante :

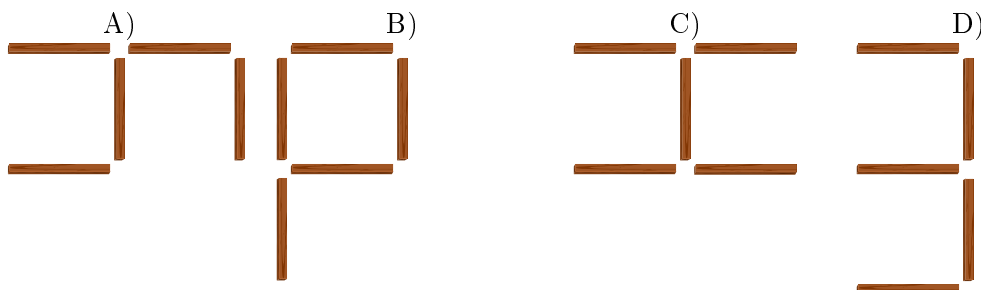


Nola prend un petit bout de bois et le déplace comme suit :



Ensuite, Bernard prend un autre petit bout de bois et le déplace aussi.

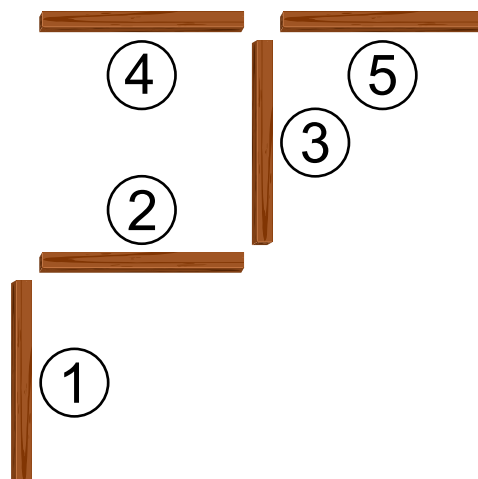
Lequel des arrangements de petits bouts de bois suivants ne correspond pas à un arrangement possible après la dernière opération ?





Solution

Pour expliquer la solution, nous numérotons les petits bouts de bois. Après l'opération effectuée par Nola, les petits bouts de bois sont arrangés comme suit :



Pour que les petits bouts de bois soient arrangés comme illustré dans la figure A), Bernard doit déplacer le petit bout de bois n^o 1 à un autre endroit.

Pour que les petits bouts de bois soient arrangés comme illustré dans la figure B), Bernard doit déplacer le petit bout de bois n^o 5 à un autre endroit.

Pour que les petits bouts de bois soient arrangés comme illustré dans la figure C), Bernard doit déplacer le petit bout de bois n^o 1 à un autre endroit.

Pour que les petits bouts de bois soient arrangés comme illustré dans la figure D), Bernard devrait déplacer deux petits bouts de bois, les petits bouts de bois n^o 1 et 5. Le problème, c'est qu'il ne peut déplacer qu'un seul petit bout de bois.

C'est de l'informatique !

Pour changer la position des petits bouts de bois, Nola et Bernard ont effectué une opération simple : ils ont choisi un seul petit bout de bois et l'on déplacé à un endroit de leur choix. Ils suivent ainsi une instructions simple qui est : «Prends un petit bout de bois quelconque et déplace-le.» Cependant, cette instruction n'est pas tout à fait évidente car ce n'est pas clairement indiqué lequel des petits bouts de bois il faudra déplacer et, par conséquent, nous ne saurons pas à coup sûr lequel des arrangements sera l'arrangement final.

Quand on programme des ordinateurs, on doit se servir d'instructions précises pour que l'ordinateur les comprenne sans équivoque. L'ordinateur est déjà bien familiarisé avec certaines instructions. Quand on veut en créer de nouvelles, on peut procéder comme suit : on peut, par exemple, créer des instructions plus complexes en concaténant différentes instructions. On peut aussi créer un programme qui exécute une seule et même instruction à plusieurs reprises ou on peut définir qu'une instruction ne soit exécutée que dans des conditions particulières. Voilà les trois possibilités les plus importantes afin de transformer des instructions simples en des instructions plus complexes.

Sites web et mots clés

instructions, changement d'état (informatique), programmer

— [https://en.wikipedia.org/wiki/Statement_\(computer_science\)](https://en.wikipedia.org/wiki/Statement_(computer_science))



- https://fr.wikipedia.org/wiki/Programmation_informatique
- https://fr.wikipedia.org/wiki/Diagramme_états-transitions





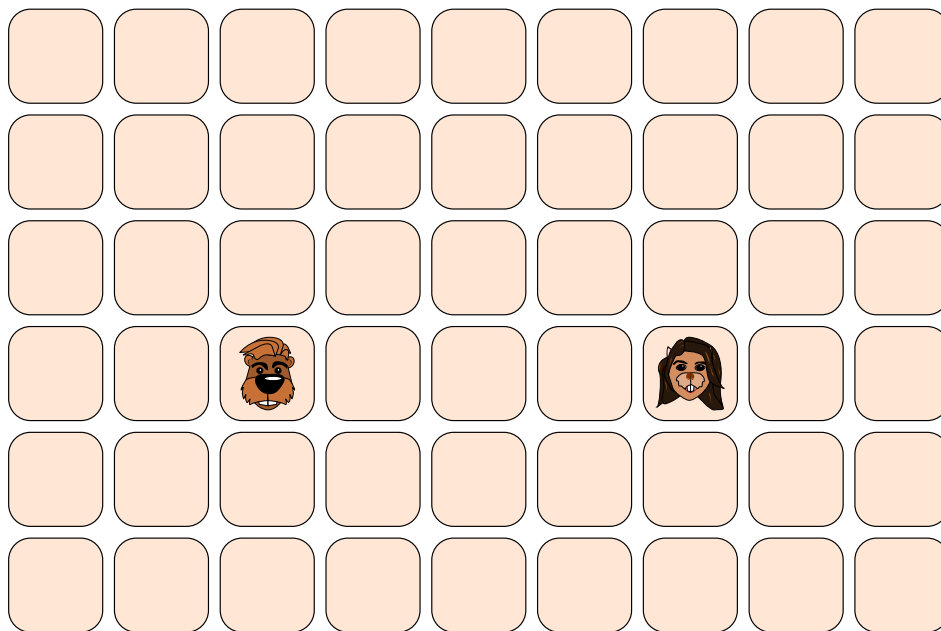
11. Concours de danse : à qui le tour ?

Un écureuil (🐿️) et un castor (🦫) participent à un spectacle de danse. Or, leurs mouvements dépendent des spectateurs : selon leurs réactions, les participants effectuent différents pas de danse. Le tableau suivant montre quelle réaction déclenche quels mouvements :

	Waouh !	Aïe !	Applaudissements !	Huées !
🐿️	← ↑	↑ ←	← ← ↑	↓ ↓
🦫	↑ →	→ ↓	↑ ↑ ↑	← ←

Quand, par exemple, les spectateurs crient «Aïe!», l'écureuil se déplacera d'abord d'une case vers le haut, puis d'une case vers la gauche ; en même temps, le castor se déplacera d'abord d'une case vers la droite, puis d'une case vers le bas.

Les deux participants commencent leur spectacle à partir des points de départ suivants :



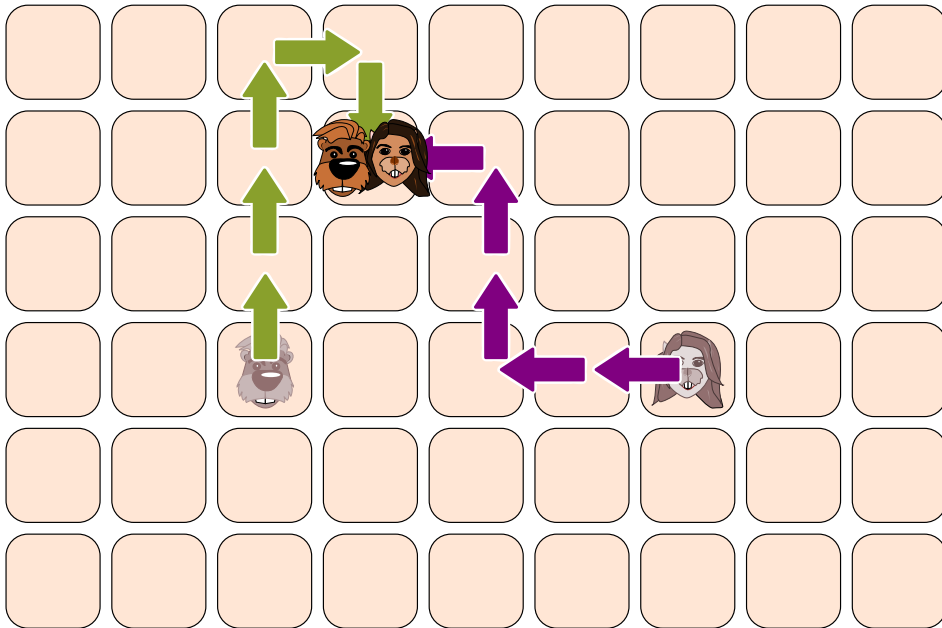
Laquelle des séquences de réactions suivantes provoquera une rencontre inévitable des deux participants sur la même case ?

- A) Huées! Aïe!
- B) Waouh! Aïe!
- C) Aïe! Aïe!
- D) Applaudissements! Aïe!



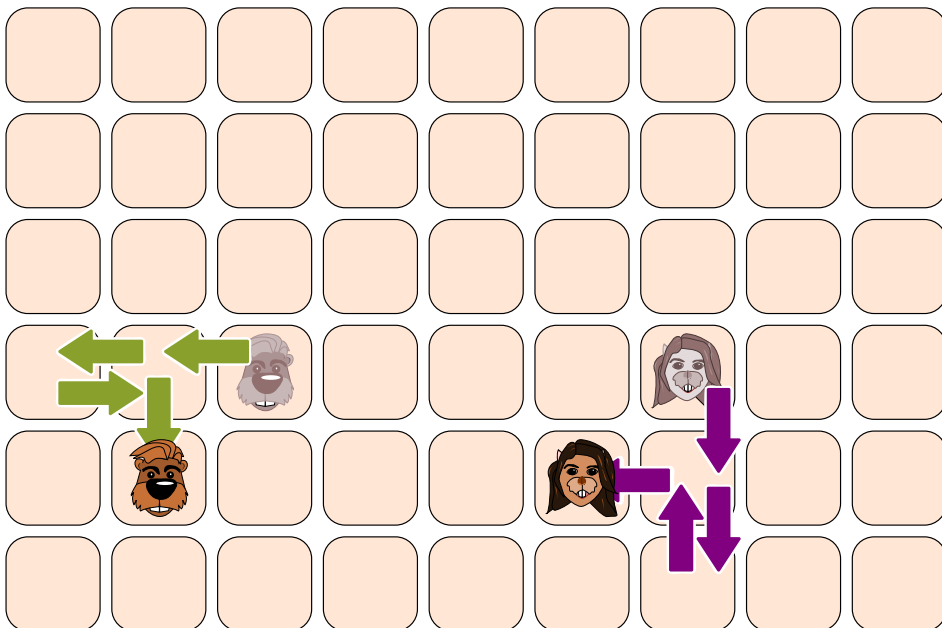
Solution

La réponse D) «Applaudissements! Aïe!» est correcte :



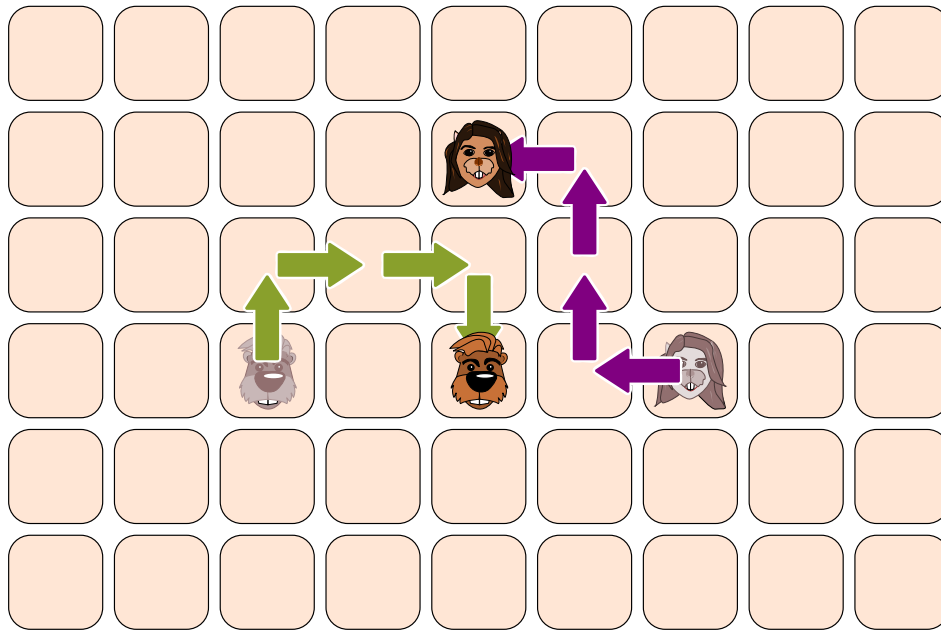
Dans tous les autres cas, l'écureuil et le castor ne se rencontreront pas sur la même case :

A) «Huées! Aïe!» :

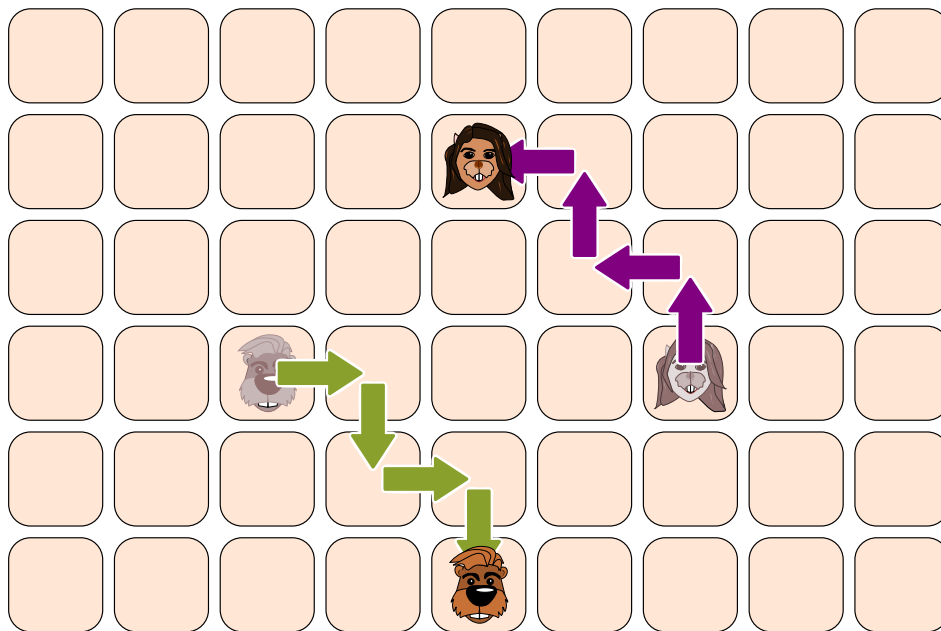




B) «Waouh! Aïe!» :



C) «Aïe! Aïe!» :



C'est de l'informatique !

La présente tâche illustre le *parallélisme* (informatique) c'est-à-dire un processus informatique qui permet de traiter des informations de manière simultanée afin de réaliser le plus grand nombre d'opérations dans un temps minimal. Dans notre cas, les deux «acteurs» bougent de manière indépendante tout en respectant des règles prédéfinies. Cependant, en informatique, au cas où deux acteurs entrent en collision, par exemple, quand ils font appel à la même ressource et ceci en même temps (une ressource peut être la mémoire vive ou un appareil raccordé à l'ordinateur comme par exemple une imprimante), il est nécessaire de préciser lequel des deux acteurs pourra accéder à la ressource en premier. Ceci est possible à l'aide des *sémaphores*, par exemple. Ainsi, un acteur peut



réserver et utiliser des ressources pour les libérer ensuite quand il n'en aura plus besoin. Seule règle : pendant qu'un acteur a réservé une ressource, aucun des autres acteurs n'a le droit d'accéder à la même ressource. Pour éviter que deux acteurs réservent une seule et même ressource, il faut donc un système central (en général, il s'agit du système d'exploitation), à l'intérieur duquel l'*ordonnanceur* gère les réservations des ressources et choisit l'ordre d'exécution des processus sur les processeurs d'un ordinateur.

Pendant que tu réfléchissais à la résolution du problème, tu as probablement effectué une *simulation* des mouvements des deux acteurs. Les *simulations informatiques* peuvent se révéler très pratiques au quotidien aussi. Ainsi, les prévisions météorologiques, par exemple, sont élaborées à l'aide des modèles numériques et de l'analyse des résultats effectuée par les prévisionnistes.

Sites web et mots clés

parallélisme (informatique), sémaphore, ordonnanceur, simulation informatique

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



12. Le nom japonais

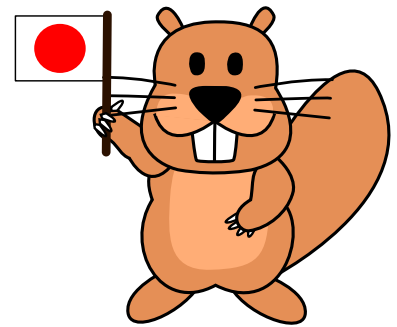
Une amie japonaise nous raconte que, selon une ancienne tradition, on peut transformer les lettres d'un prénom en un nom spirituel japonais. Il faut simplement remplacer chaque lettre par une syllabe qui lui est attribuée :

A → ka	F → lu	K → me	P → mor	U → do	Z → zi
B → pi	G → ji	L → ta	Q → ke	V → ru	
C → mi	H → ri	M → rin	R → shi	W → mei	
D → te	I → ki	N → to	S → ari	X → na	
E → ku	J → zu	O → mo	T → chi	Y → fu	

Un de ses amis provenant de la Croatie porte, par exemple, le nom spirituel «Zukame Moru».

Quel est le vrai nom de son ami croate ?

- A) Josip
- B) Jani
- C) Jakov
- D) Jurica





Solution

Son ami s'appelle Jakov. Pour trouver la bonne solution, on peut débiter par la syllabe «Zu» qui, selon le tableau, a été attribuée à la lettre «J». Après avoir trouvé, de la même manière, la lettre qui correspond à la syllabe «ka» (la lettre «A»), on réalise que les deux solutions «Josip» et «Jurica» ne sont pas correctes. De même, après avoir trouvé la lettre originale à laquelle est attribuée la syllabe «me» (la lettre «K»), on comprend que la solution C) est correcte. Pour être sûr, on peut vérifier le mot «Moru» en traduisant directement les lettres «O» («mo») et «V» («ru») en syllabes. Quand on traduit les lettres en syllabes, on peut procéder plus vite qu'à l'inverse car les lettres du tableau suivent l'ordre alphabétique, les syllabes, elles, par contre, ne sont pas triées.

Une autre approche, plus vite encore, serait de chercher dès le début la lettre originale qui correspond à la syllabe «ru». Celle-ci remplace la lettre «V» et comme Jakov est le seul nom qui se termine par un «V», on peut être sûr qu'il s'agit du prénom que l'on cherche.

C'est de l'informatique !

Il se peut que tu aies déjà entendu parler des jeux similaires. Souvent, ils sont utilisés pour créer des langages secrets. L'idée de base repose sur le fait qu'une lettre est substituée à une autre lettre ou à une syllabe distinctive. En informatique, ce langage est appelée *langage congruentiel* ou *système de réécriture* ou encore *système de semi-Thue*.

Malheureusement, ces méthodes de chiffrement ne sont pas sûres car elles ont pour base des *algorithmes de chiffrement monoalphabétiques*. Elles sont ainsi vulnérables aux attaques et peuvent être facilement déchiffrées, même sans que l'on ait recours à un ordinateur.

Notre tâche révèle un autre défi au niveau de l'informatique : dans la partie «Moru», on peut également identifier la syllabe «mor» qui remplace la lettre «P». La syllabe «mo» qui, elle, remplace la lettre «O», est en fait un préfixe de «mor». Au cas où un ordinateur doit déchiffrer un prénom spirituel, l'informaticien doit être conscient du problème des préfixes afin que ses réflexions ne conduisent pas à des conclusions erronées.

D'ailleurs, toutes les syllabes que tu trouves dans le tableau de notre tâche proviennent du langage japonais. C'est la raison pour laquelle tous les prénoms traduits selon cette méthode sonnent fortement comme de vrais noms japonais.

Sites web et mots clés

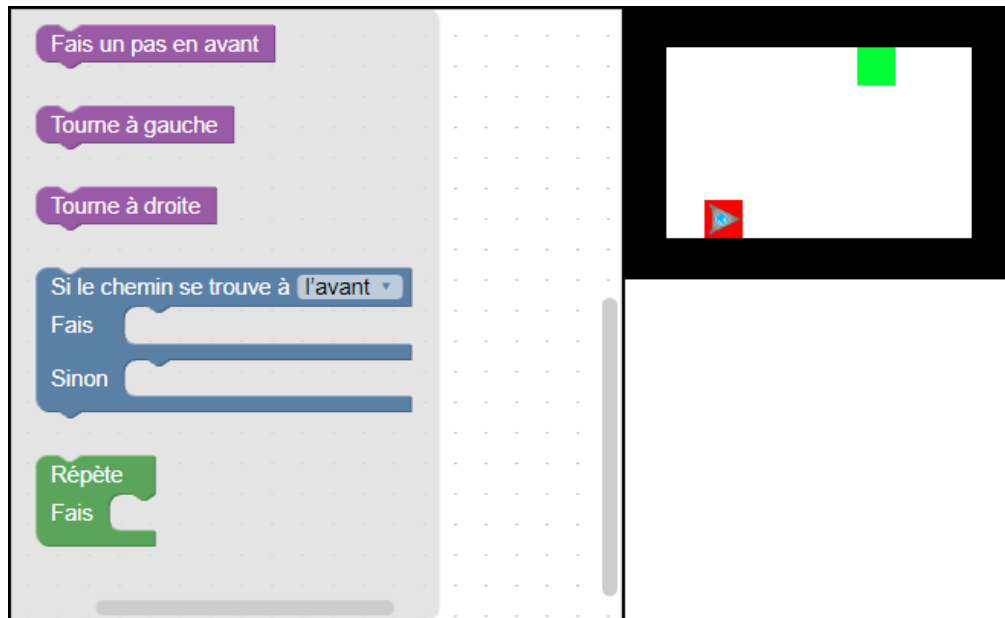
langage congruentiel, système de réécriture, système de semi-Thue, chiffrement monoalphabétique.

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



13. Un programme court

Un robot de forme triangulaire doit parcourir le chemin suivant : il commence tout en bas dans la zone marquée en rouge et se dirige vers la cible, qui est la zone marquée en vert. Malheureusement, il n'arrive à lire que des programmes très courts.

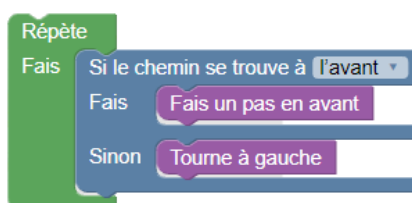


Pour créer un tel programme, déplace les instructions et insère-les dans l'organigramme de programmation afin qu'elles suivent un ordre logique. Seule règle : tu ne dois pas utiliser plus de 4 instructions.



Solution

Voilà une solution possible :



L'idée de base de ce programme est de faire avancer le robot le long du mur et de le faire tourner à gauche à chaque angle. Ainsi, il arrivera automatiquement à destination.

C'est de l'informatique !

Dans le champ formé de 8×5 pas, le robot devrait au moins effectuer neuf mouvements pour arriver à destination : tout droit – tout droit – tout droit – tout droit – à gauche – tout droit – tout droit – tout droit – tout droit.

Pourtant, le programme ne comprend que quatre instructions, c'est la raison pour laquelle nous avons besoin d'une boucle (loop) qui permettra au robot de faire autant de pas en avant et autant de virages à gauche que nécessaire jusqu'à ce qu'il arrive à destination. En outre, pour pouvoir contrôler les mouvements du robot, surtout celui où il faut tourner à gauche sans que le robot entre en collision avec le mur, il faut introduire une instruction conditionnelle.

Malheureusement, ce programme de courte durée ne permet pas de trouver le chemin le plus court et puis, il ne fonctionne pas dans toutes les situations non plus. Si, par exemple, notre cible ne se trouvait pas le long d'un mur, le robot ne la trouverait en aucun cas.

Les séquences d'instructions, les boucles ainsi que les instructions conditionnelles constituent la base de chaque algorithme.

Sites web et mots clés

programmer, boucle, condition, instruction

- <https://fr.wikipedia.org/wiki/Blockly>
- <http://primalogo.ch/fr>
- <http://www.abz.inf.ethz.ch/primarschulen-stufe-sek-1/programmieren-fur-kinder/>
- http://www.swisseduc.ch/informatik/programmiersprachen/scratch_werkstatt/
- <http://ilearnit.ch/de/2b/explain.html>
- <https://scratch.mit.edu/>
- <http://kinderlabor.ch/informatik-fuer-kinder/programmieren-mit-scratch/>

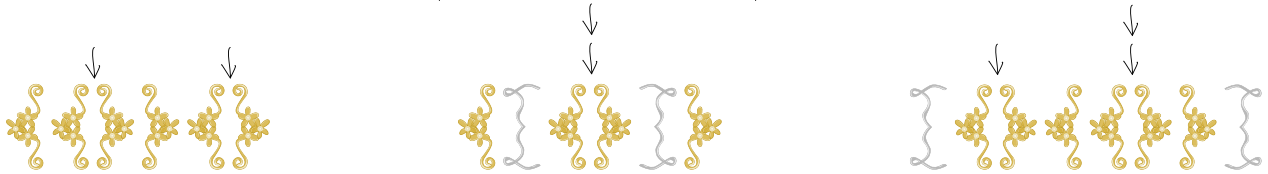


14. Orner des pendentifs médiévaux

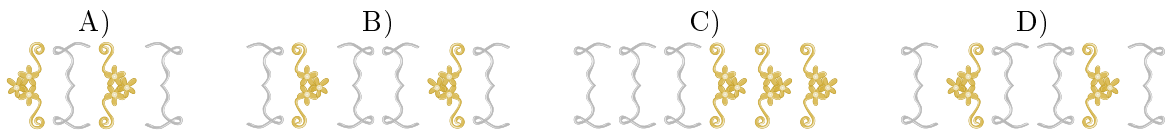
Pour une fête médiévale, la famille Castor produit des bijoux, plus précisément des pendentifs. La décoration de ces derniers consiste à placer des ornements qui ressemblent à des accolades. Chaque accolade est utilisée par paires. Pour décorer le bijou, les Castor appliquent une méthode particulière : on commence par une des deux paires d'accolades suivantes :



Ensuite, on ajoute une autre paire d'accolades à plusieurs reprises et à un endroit quelconque, à la seule condition que les paires supplémentaires soient placées entre deux accolades d'une paire d'accolades placée préalablement (voir l'exemple ci-dessous) :



Lequel des pendentifs a été décoré selon la méthode décrite ci-dessus ?





Solution

La réponse D) est correcte. Entre les deux accolades de l'ornement initial, on a placé une paire d'accolades supplémentaire, entre les accolades de cette dernière, on a placé une autre paire d'accolades. Tous les autres pendentifs n'ont pas été décorés selon la méthode développée par la famille Castor :

- A) Partant de la gauche, la première erreur s'est produite à la position 3 : on a fermé la première paire d'accolades avant que l'on ait pu placer la seconde accolade de la deuxième paire (position 2).
- B) Partant de la gauche, la première erreur s'est produite à la position 1 : on a placé la seconde accolade de l'ornement initial sans avoir placé la première accolade.
- C) Partant de la gauche, la première erreur s'est produite à la position 4 : on a fermé une paire d'accolades avant même de l'avoir ouverte.

C'est de l'informatique !

Les règles de l'ornement des pendentifs correspondent tout à fait à celles utilisées en mathématiques ou en informatique. Si une entité lexicale d'un langage informatique est correcte, on l'appelle «bien formée». Une entité bien formée est également appelée *syntactiquement correcte* parce qu'elle respecte la grammaire formelle du langage en question, c'est-à-dire les règles d'agencement des entités lexicales de ce langage. En règle générale, il est plus facile de corriger une suite d'entités lexicales d'un langage qui comporte des erreurs de syntaxe qu'une suite d'entités qui comporte des «erreurs sémantiques». Ces dernières sont plus subtiles et peuvent comprendre des erreurs de raisonnement.

Sites web et mots clés

bien formée, syntaxe, sémantique

- <https://fr.wikipedia.org/wiki/Syntaxe>
- <https://fr.wikipedia.org/wiki/Sémantique>



15. Le journal scolaire

L'équipe de rédaction du journal scolaire comprend 10 volontaires. Tous les vendredis, ils travaillent pendant leurs heures creuses pour rédiger des articles ou des reportages. L'emploi du temps ci-dessous nous montre les heures creuses individuelles des volontaires (en vert) pendant lesquelles chacun d'entre eux travaille pour le journal scolaire :

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Anna								
Bea								
Celine								
David								
Emma								
Flo								
Gioa								
Hans								
Ida								
Jakob								

Comme les volontaires travaillent très bien, le proviseur a décidé de mettre à leur disposition de nouveaux ordinateurs portables.

Combien de nouveaux ordinateurs portables l'école doit-elle acquérir pour que chaque volontaire puisse travailler sur un de ces nouveaux ordinateurs pendant ses heures creuses individuelles ?

- A) quatre
- B) cinq
- C) sept
- D) dix



Solution

La réponse A) n'est pas correcte. Entre 9h et 10h, les volontaires 1, 3, 5, 8 et 9 devraient travailler parallèlement. Ainsi, quatre ordinateurs portables ne seraient pas suffisants.

La réponse B) est correcte. De 9h à 10h et de 10h à 11h, il y a exactement 5 volontaires qui travaillent parallèlement. 5 est aussi le nombre maximum de volontaires qui travaillent en même temps. Ceci veut dire que pendant toutes les autres heures, il y a toujours moins de 5 volontaires qui travaillent en même temps. Il faut donc acquérir 5 ordinateurs portables au maximum.

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Anna								
Bea								
Celine								
David								
Emma								
Flo								
Gioa								
Hans								
Ida								
Jakob								

Étant donné qu'il ne faut que 5 ordinateurs portables au maximum, les réponses C) et D) ne sont pas correctes.

C'est de l'informatique !

Afin de mieux comprendre les interdépendances à l'intérieur d'une grande quantité de données, il s'avère utile de choisir une représentation des données appropriée comme, par exemple, un tableau, un diagramme ou un graphe. Selon le but visé, les différentes représentations sont plus ou moins avantageuses.

Pour notre tâche, il conviendrait de représenter les données comme suit :

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
	Celine	Celine	Bea	Bea	Bea	Bea		
		Emma	Emma	Flo	Flo	Jakob	Jakob	
		Anna	Anna		David	David	David	
	Ida	Ida	Ida					
		Hans	Gioa	Gioa		Gioa	Gioa	

Cette représentation permet d'associer un ordinateur portable à l'heure ou aux heures de travail individuelles de chaque volontaire. Ainsi, on peut rapidement vérifier qu'il faut acquérir 5 ordinateurs portables au maximum.

Dans notre quotidien, on a souvent besoin de telles planifications surtout quand il s'agit d'exploiter les ressources disponibles de manière optimale. Chaque hôtel, par exemple, possède un système de réservation. D'une part, il est important que les réservations ne se recoupent pas, et d'autre part, il est souhaitable que l'hôtel affiche complet. Il en va de même pour les salles de classe d'une école. Là aussi, une planification optimale est souhaitable pour tous les acteurs concernés, même si le terme «optimal» peut être compris de manières diverses : serait-il par exemple préférable qu'une classe reste toute la journée dans la même salle ou, qu'au contraire, les professeurs ne changent pas de salle de classe ?



Il existe encore un autre type de représentation pour de telles planifications : c'est le graphe d'intervalles. Chaque intervalle de ce graphe représente l'allocation d'une ressource pendant un certain temps et les arêtes relient deux sommets lorsque deux intervalles correspondants se recoupent.

Sites web et mots clés

planification, graphe d'intervalles

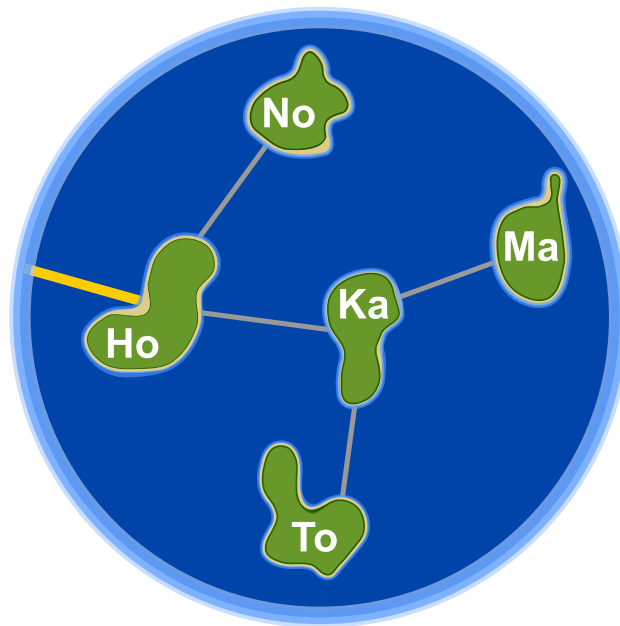
— https://fr.wikipedia.org/wiki/Graphe_d'intervalles





16. Honomakato

L'archipel Honomakato est formé de cinq îles Ho, No, Ma, Ka et To. L'île principale Ho est connectée à Internet par un câble. En outre, quelques câbles parcourent les îles Ho et No, Ho et Ka, Ka et Ma ainsi que Ka et To. Toutes les îles sont donc connectées à l'île principale Ho et par conséquent à Internet.



Les habitants de Honomakato demandent une connexion fiable à Internet pour toutes les îles : cela veut dire que même si un des câbles est endommagé, chacune des îles doit avoir accès à Internet. *Fais en sorte que l'archipel Honomakato obtienne une connexion fiable à Internet. Pose deux autres câbles entre les îles. Il existe plusieurs solutions possibles.*

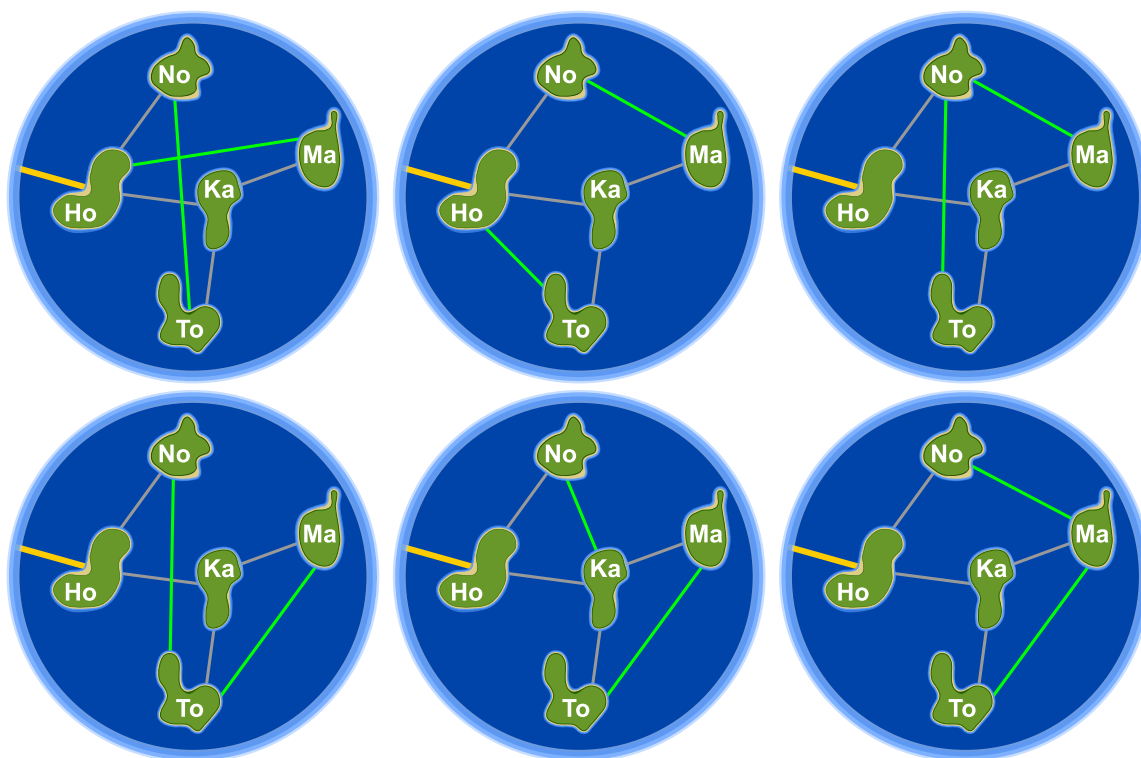


Solution

En posant deux autres câbles, l'archipel Honomakato obtient une connexion fiable à Internet. Pour résoudre notre problème, il existe six possibilités différentes. Les câbles posés préservent les îles du moment inopportun où elles n'auraient plus accès à Internet si un câble était endommagé.

1. Ho-Ma et No-To : Ho-Ma préserve Ma und Ka, No-To préserve No und To.
2. Ho-To et No-Ma : Ho-To préserve To et Ka, No-Ma préserve Ma, No, et Ka.
3. No-To et No-Ma : No-To préserve No, To et Ka, No-Ma préserve Ma, No, et Ka.
4. No-To et Ma-To : No-To préserve No, To et Ka, Ma-To préserve Ma et To.
5. No-Ka et Ma-To : No-Ka préserve No et Ka, Ma-To préserve Ma et To.
6. No-Ma et Ma-To : No-Ma préserve Ma, No, et Ka, Ma-To préserve Ma et To.

En général, pour chaque solution, les règles suivantes doivent être respectées : (1) chaque île est équipée d'au moins deux connexions et (2) il n'est pas possible de diviser l'archipel Honomakato en deux groupes qui n'ont qu'une seule connexion.



C'est de l'informatique !

D'une part, le réseau de câbles avec lequel l'archipel Honomakato est connecté à Internet ne représente qu'une petite partie du réseau global. D'autre part, il sert aussi d'exemple pour montrer comment le réseau global est construit. Les routeurs, les serveurs et les autres dispositifs télématiques pourvus d'une propre adresse Internet sont des nœuds d'un grand réseau global appelé «Internet» ; dans notre tâche, les îles de l'archipel Honomakato représentent justement ces nœuds.

Internet a été conçu dans les années 1960 comme un réseau robuste (appelé aussi «fiable»). Un des objectifs était qu'une panne de connexion entre les nœuds de réseau ne provoquerait pas une panne dans le réseau entier. C'est la raison pour laquelle on connecte les nœuds de manière multiple et on les



configure de façon à ce que, en cas de défaillance ou congestion d'une connexion, on ait la possibilité de recourir à une autre connexion. Cette précaution est également importante pour d'autres réseaux, comme par exemple pour les réseaux de transport ou les réseaux d'approvisionnement.

En informatique, on utilise la théorie des graphes pour effectuer des calculs concernant ces types de réseaux. Un graphe (en théorie des graphes) est un réseau qui se compose de «points» appelés des *nœuds* ou *sommets* et de «liens» entre les nœuds appelés des *arêtes*. Un graphe est appelé «connexe» lorsque pour chaque paire de nœuds A et B, B est connecté avec A à travers au moins une arête. Une seule arête est donc nécessaire pour qu'un graphe soit connecté et dans ce cas-là, l'arête assume la fonction d'un *pont*. En informatique, on développe des algorithmes qui sont capables de repérer ces ponts à l'intérieur des graphes. Robert Tarjan a développé un de ces algorithmes (efficaces).

Sites web et mots clés

structure de données dynamique, graphe, pont (informatique)

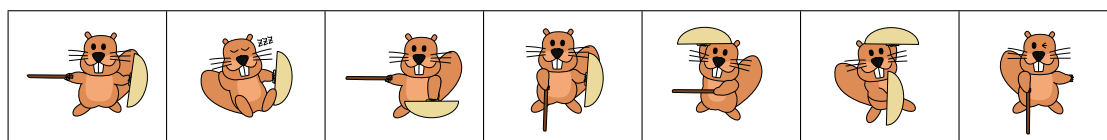
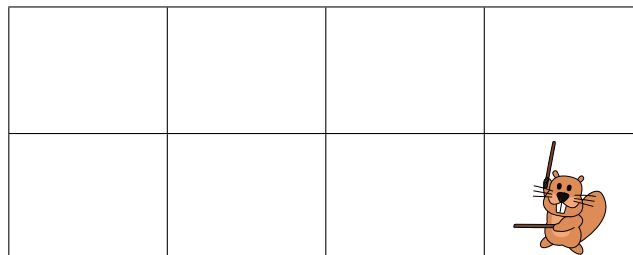
— [https://fr.wikipedia.org/wiki/Séparateur_\(théorie_des_graphes\)](https://fr.wikipedia.org/wiki/Séparateur_(théorie_des_graphes))





17. Un art martial japonais

Lucia et ses amis sont membres d'un club d'art martial japonais qui enseigne le maniement du bâton. Pour une photo dans la cour de récréation, ils aimeraient bien se mettre en place afin que chaque bâton vise un bouclier. Pour que chacun puisse se mettre correctement en place, on a dessiné quelques cases sur le sol de la cour de récréation. Lucia a déjà choisi une case et elle montre sa pose préférée. En dessous, tu peux voir tous ses amis qui présentent leur propre pose préférée :

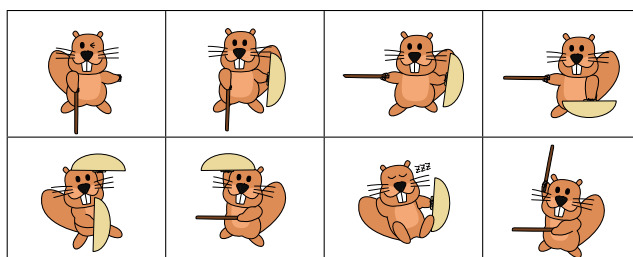


Déplace les images des amis dans les cases dessinées sur le sol de la cour de récréation pour que chaque bâton vise un bouclier.



Solution

La réponse correcte est :



Si on déplace les images des amis comme indiqué ci-dessus, chaque bâton visera un bouclier. Pour résoudre ce problème, il n'existe pas d'autre solution.

C'est de l'informatique !

Nous avons sept images qu'il faut déplacer et mettre au bon endroit. Celui qui tente de résoudre la tâche par essai-erreur aura besoin de beaucoup de temps : il existe en fait $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 = 7! = 5040$ arrangements possibles et la plus grande partie d'entre eux est naturellement fausse. Par la logique, tu pourras trouver la solution plus rapidement :

1. Tous les castors dont le bâton ou le bouclier montre le haut doivent être placés dans la rangée inférieure.
2. Tous les castors dont le bâton ou le bouclier montre le bas doivent être placés dans la rangée supérieure.
3. Il n'y a qu'un seul castor dont le bouclier montre vers le bas. C'est la raison pour laquelle il doit être placé au-dessus de Lucia.

Ces quelques règles aident à limiter l'espace de recherche des diverses solutions possibles et finalement, à trouver la bonne. Le fait de tester systématiquement toutes les options possibles par essai-erreur est appelé le *backtracking*. Mais une telle méthode n'est raisonnable que si l'on a restreint l'espace de recherche auparavant. Voilà pourquoi la délimitation de l'espace de recherche par des règles logique est si importante.

Sites web et mots clés

raisonnement logique, conclusion

- https://fr.wikipedia.org/wiki/Retour_sur_trace
- http://www.inf-schule.de/grenzen/komplexitaet/affenpuzzle/einstieg_affenpuzzle



18. La confiture de grand-père

Anna, Pierre et Lisa aident leur grand-père à mettre de la confiture en pots. Voilà les étapes de travail qu'il faut faire – et ceci impérativement dans l'ordre suivant :



Rincer un pot de confiture prend 3 minutes.



Mettre de la confiture dans un pot prend 2 minutes.



Fermer un pot de confiture prend 1 minute.

Anna, Pierre et Lisa aimeraient bien se répartir ces différentes étapes de travail et pour cela, ils établissent un plan. Ils doivent respecter les règles suivantes : une tâche doit être complètement accomplie avant qu'ils n'entament la prochaine. Ainsi, un pot de confiture ne peut être fermé avant qu'il ait été d'abord rincé, puis rempli de confiture.

Ce plan, par exemple, ne respecte pas les règles préalablement déterminées :



ANNA										
PIERRE										
LISA										

Anna, Pierre et Lisa aimeraient bien remplir un maximum de pots en 10 minutes. Aide-les à établir un plan qui respecte toutes les règles.

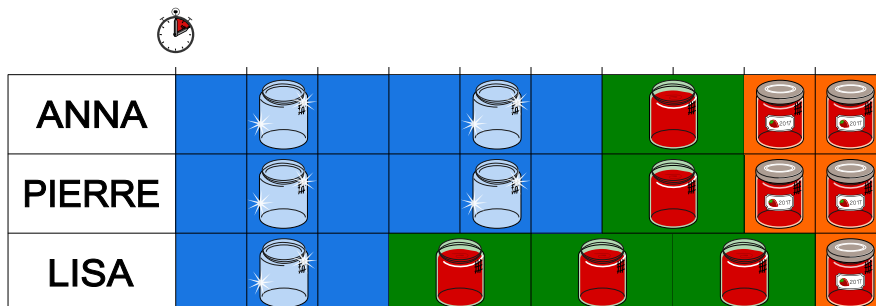


ANNA										
PIERRE										
LISA										



Solution

Voilà un plan selon lequel Anna, Pierre et Lisa arrivent à mettre de la confiture en pots en 10 minutes. Il existe naturellement bien d'autres plans selon lesquels ils arriveraient à atteindre leur objectif. Par exemple, ils pourraient tout simplement conserver les mêmes agencements de tâches, mais en se les répartissant différemment entre eux.



Mais ils n'arriveront jamais à remplir plus de 5 pots en 10 minutes. Remplir un pot prend 6 minutes ($3 + 2 + 1 = 6$). Chacun des trois enfants ne peut travailler que pendant 10 minutes. Le temps de travail au total est donc de 30 minutes. Pendant ces 30 minutes, ils arrivent à remplir $30 : 6 = 5$ pots au maximum.

C'est de l'informatique !

Dans cette tâche, les trois enfants Anna, Pierre et Lisa travaillent en même temps, c'est-à-dire qu'ils effectuent en parallèle les différentes étapes de travail. En informatique, le traitement en parallèle est un domaine important. Les centres de calculs traitent à l'aide de beaucoup d'ordinateurs de grandes quantités de données en parallèle. Pour ce faire, on doit distribuer les différentes opérations à différents ordinateurs de manière logique, comme c'était le cas dans notre tâche quand il s'agissait de répartir les différentes étapes de travail aux trois enfants. Les ordinateurs individuels ou les smartphones, eux aussi, possèdent plusieurs processeurs principaux (multiprocesseurs) capables de traitements parallèles. En informatique, on appelle la planification du travail *scheduling*.

En dehors de l'informatique, le scheduling est très fréquent aussi. Dans la gestion de grands projets, par exemple, pour planifier les différentes étapes, on a souvent recours à la *méthode des antécédents*, appelée aussi *réseau des antécédents*. Cet outil de planification de projets détermine l'ordre des tâches ainsi que leur durée nécessaire. La mise en oeuvre d'un plan de réseau fonctionne comme le scheduling. Ainsi, les connaissances en informatique peuvent toujours être utiles, même si on ne travaille pas comme informaticienne ou informaticien.

Sites web et mots clés

scheduling, traitement parallèle, méthode des antécédents

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

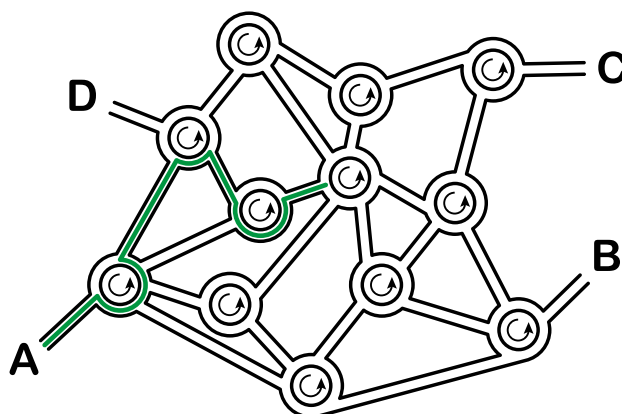


19. La ville riche en ronds-points

Dans la ville des castors, tous les carrefours ont la forme de ronds-points. Quand les habitants de la ville expliquent le chemin à un touriste, ils lui disent simplement :

- Au prochain ronds-point, prends la quatrième sortie.
- Au prochain ronds-point, prends la première sortie.
- Au prochain ronds-point, prends la deuxième sortie.

Si la personne connaît déjà assez bien la ville mais qu'elle cherche un endroit particulier, les castors ne lui indiquent qu'une suite de chiffres comme par exemple «4 1 2». Cette personne comprendra donc tout de suite qu'il faudra prendre l'itinéraire suivant :



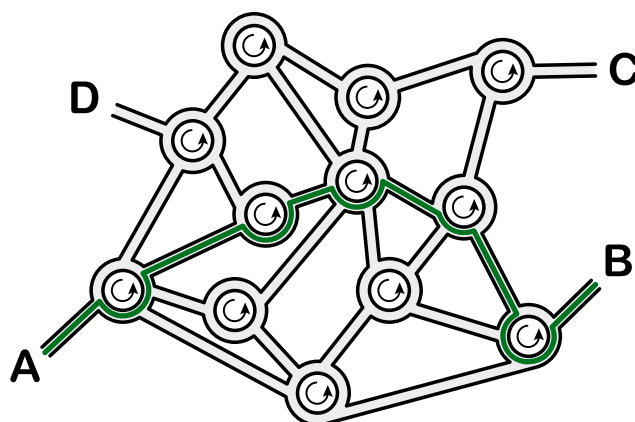
Quand un touriste part du point A, à quel endroit l'indication «3 1 3 2 3» le mènera-t-elle ?

- A) L'indication le mènera au point A.
- B) L'indication le mènera au point B.
- C) L'indication le mènera au point C.
- D) L'indication le mènera au point D.



Solution

L'indication «3 1 3 2 3» le mènera au point B :



C'est de l'informatique !

Cette tâche est un bon exemple pour illustrer «l'information structurée». Un ordinateur comprend difficilement un itinéraire formulé en détail dans notre langue. En transformant notre langue, par exemple, en une suite de chiffres, comme c'est le cas dans notre exercice, l'ordinateur sera tout de suite capable d'interpréter ces informations comme une suite d'instructions en langage machine. On appelle une telle suite une *séquence*. La séquence d'instructions est à la base de beaucoup de langages de programmation.

Dans notre exemple, il est utile que le réseau routier soit uniforme : tous les carrefours se présentent sous forme de ronds-points. On appelle de telles structures uniformes «homogènes» contrairement à des structures variées appelées «hétérogènes». En informatique, on préfère les structures homogènes aux structures hétérogènes car elles peuvent être traitées de manière automatique et efficace par un logiciel, donc beaucoup plus vite que les structures hétérogènes.

Sites web et mots clés

séquences, exécution de programmes, langage formel

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



20. Pizzeria Castoria

La Pizzeria Castoria n'a qu'un seul four à pizza. C'est pourquoi le pizzaiolo ne peut cuire que peu de plats en même temps.

En tout, il y a trois combinaisons possibles :



Le temps de cuisson varie selon le type de repas : une petite pizza doit cuire pendant 10 minutes, une grande pizza pendant 15 minutes et la cuisson d'un pain ciabatta, par contre, nécessite 20 minutes. Ce qui facilite un peu la tâche du pizzaiolo est que même s'il est en train de cuire deux ou trois plats en même temps, il peut les mettre au four ou les sortir du four au bon moment, c'est-à-dire il peut les y mettre ou les en sortir successivement.

Aujourd'hui, le pizzaiolo a beaucoup de travail. Il doit préparer une petite pizza, deux grandes pizzas et quatre pains ciabatta. Les clients sont affamés et ils attendent leurs plats commandés le plus vite possible.

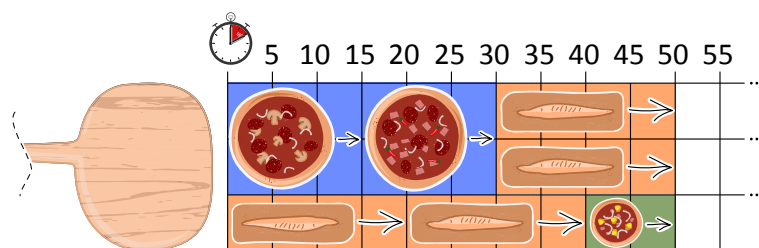
Quel est le temps minimal pendant lequel le pizzaiolo arrive à faire cuire tous les plats commandés ?





Solution

Pour résoudre ce problème, il existe plusieurs solutions optimales. Une d'entre elles serait de cuire d'abord les deux grandes pizzas successivement tout en cuisant parallèlement deux pains ciabatta. Puis on mettra au four deux pains ciabatta et une petite pizza :

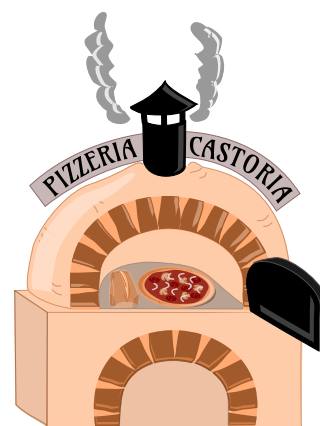


Ainsi, le temps de cuisson serait de 50 minutes au total. Il n'existe pas de cuisson plus rapide car avec cette solution, le four est chargé de manière maximale pendant tout le temps de la cuisson.

C'est de l'informatique !

Quand on essaie de planifier l'ordre d'exécution des processus, dit d'ordonnancement (en anglais, *scheduling*), et ainsi de planifier les demandes d'accès à une ressource limitée (dans notre tâche, il s'agit du four ou plus exactement de la charge maximale du four), on cherche à trouver une solution optimale en ce qui concerne les coûts ou le temps d'exécution (voir notre problème à résoudre). Si la ressource est utilisée de manière optimale, le temps d'exécution des processus sera minimal.

Un algorithme d'ordonnancement très populaire est le *round-robin* (en anglais, *Round-Robin* qui, lui, est en fait un jeu de parcs, un tourniquet). L'idée de cet algorithme est que tous les processus qui doivent être exécutés sont insérés dans une file d'attente gérée comme une file circulaire : l'ordonnanceur accorde un temps processeur à chacun des processus. Si ce temps-là n'est pas suffisant, le processus sera réinséré dans la file d'attente jusqu'à ce que l'ordonnanceur lui accorde un nouvel intervalle de temps.



Cependant, le pizzaiolo de la pizzeria ne pourra pas se servir de cette stratégie car jamais il ne sortirait un plat du four à demi cuit pour l'y remettre un peu plus tard afin de terminer la cuisson. C'est pourquoi, en informatique, on recourra plutôt à une autre stratégie, appelée *greedy algorithm* (en français, *algorithme glouton* ou *gourmand*) : commençant par le processus qui demande le plus de ressources (par une grande pizza car celle-ci occupe la plus grande place dans le four), on continuera par des processus qui demandent moins de ressources et ainsi de suite. Étant donné que le four est assez grand pour faire cuire plusieurs plats en même temps (bien que ce processus soit soumis à des restrictions), le pizzaiolo peut sortir du four un plat dont la cuisson est terminée en le remplaçant par un autre plat qui est prêt à être cuit.

Pourtant, il reste à savoir si, en fin de compte, on a vraiment trouvé la solution optimale. Ceci n'est pas aussi évident car on ne peut pas exclure définitivement que l'on ne trouverait pas une meilleure solution à celle que l'on a trouvée. On pourrait, par exemple, commencer par trois pains ciabatta. Toutefois, quand on a réussi à utiliser de manière maximale toutes les ressources (dans notre tâche cela signifierait de charger le four au maximum), on peut être sûr que l'on ne trouvera pas de solution plus avantageuse.



Sites web et mots clés

scheduling, l'ordonnancement, file d'attente, ressource, round-robin, greedy algorithm, algorithme glouton

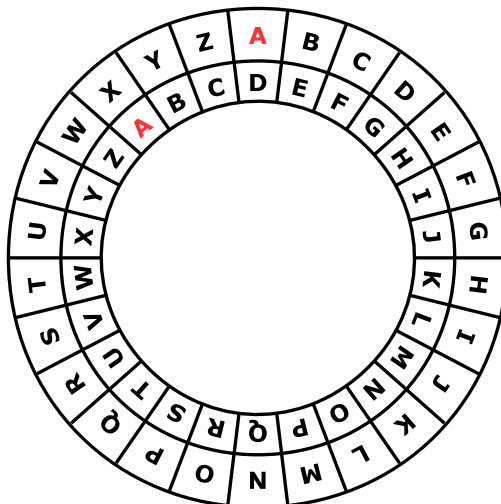
- https://fr.wikipedia.org/wiki/Ordonnancement_dans_les_systèmes_d'exploitation
- [https://fr.wikipedia.org/wiki/Round-robin_\(informatique\)](https://fr.wikipedia.org/wiki/Round-robin_(informatique))
- https://fr.wikipedia.org/wiki/Algorithme_glouton





21. Une commande chiffrée

Anna passe ses commandes au restaurant à l'aide de messages chiffrés. Seul César, le cuisinier, sait les déchiffrer. Pour rédiger un message chiffré, elle utilise un disque particulier composé d'un anneau extérieur et d'un anneau intérieur mobile. Chaque anneau affiche les lettres de l'alphabet. Celles-ci sont ordonnées dans l'ordre de l'alphabet. Au début, les lettres des deux anneaux sont alignées : la lettre A (de l'anneau intérieur) se trouve exactement en dessous de la lettre A (de l'anneau extérieur), la lettre B se trouve en dessous de la lettre B, et ainsi de suite.



Pour rédiger un message chiffré, Anna procède comme suit : d'abord, elle note sa commande, par exemple une PIZZA. Ensuite, elle effectue les opérations suivantes :

1. Au-dessous de chaque lettre du plat commandé, elle note un chiffre au hasard. Celui-ci marque le nombre de rotations qu'il faut effectuer plus tard.
2. Pour chaque lettre du message original, elle met d'abord l'anneau intérieur à la position initiale, ensuite elle le tourne dans le sens inverse des aiguilles d'une montre, lettre par lettre. Le nombre de rotations correspondra au nombre de rotations propre à la lettre du message original.
3. Finalement, elle remplace la lettre originale par la lettre que l'anneau intérieur indique au-dessous de la lettre originale.

Si, par exemple, elle veut commander une PIZZA et qu'elle utilise les nombres de rotations 3, 1, 4, 1 et 5, elle rédige le message chiffré SJDAF.

commande	P	I	Z	Z	A
nombre de rotations	3	1	4	1	5
message chiffré	S	J	D	A	F

Pour une autre commande, Anna utilise les nombres de rotations 3, 1, 4, 1, 5, 9 et 2 et ensuite, elle rédige le message chiffré OBWBLWC.

Si nous regardons de plus près le message chiffré OBWBLWC, quelle commande Anna a-t-elle passée ?

commande							
nombre de rotations	3	1	4	1	5	9	2
message chiffré	O	B	W	B	L	W	C



Solution

La réponse correcte est LASAGNA :

commande	L	A	S	A	G	N	A
nombre de rotations	3	1	4	1	5	9	2
message chiffré	O	B	W	B	L	W	C

Afin de déchiffrer le message chiffré, c'est-à-dire de savoir quelle commande Anna a passée, nous nous servons de son disque particulier. Pour trouver la lettre originale, nous tournons l'anneau intérieur selon le nombre de rotations qui correspond à cette lettre dans le sens des aiguilles d'une montre. Ensuite, on cherche la lettre chiffrée parmi les lettres de l'alphabet de l'anneau intérieur. La lettre qui se trouvera au-dessus de la lettre chiffrée sera une des lettres du message original. Donc, pour déchiffrer le message chiffré, on fait tout simplement l'inverse de ce que l'on fait quand on chiffre un message.

C'est de l'informatique !

Anna chiffre son message afin qu'il ne soit lu que par son cuisinier préféré. Le *chiffrement* ou le *cryptage* est une méthode très ancienne. Les motivations de chiffrer un message sont évidentes : nous aimerions bien être sûrs que le message ne sera lu que par un destinataire particulier et non pas par un espion, par exemple. Il existe beaucoup de procédés différents de *cryptographie*, mais, en général, nous avons affaire à deux algorithmes principaux : un pour chiffrer le message et un pour le déchiffrer. Tous les deux dépendent d'une *clé de déchiffrement* pour exécuter leur travail.

Une des méthodes de chiffrement très simple date de l'époque de Jules César : dans ce cas-là, la clé de déchiffrement est un nombre, appelé la longueur du décalage. Celui-ci indique la distance entre les deux lettres, la lettre du message original et la lettre qui servira comme lettre chiffrée. La clé de déchiffrement 3, par exemple, signifiera un décalage de 3 vers la droite dans l'ordre de l'alphabet : ainsi, la lettre A du message original sera chiffrée par la lettre D, la lettre B du message original, par contre, sera chiffrée par la lettre E, etc. En même temps, la clé de déchiffrement indique l'inverse, notamment que la lettre D a substitué la lettre A et la lettre E a substitué la lettre B. Cette méthode de chiffrement est appelée *chiffrement par décalage*, aussi connu comme le *chiffre de César*. Tout comme dans notre tâche, cette méthode fonctionne à l'aide d'une superposition de deux alphabets.

Des méthodes de chiffrement qui ont pour base une seule clé de déchiffrement (par exemple une seule longueur de décalage) sont comparativement peu sûres. Apparemment, Anna en est parfaitement consciente car elle utilise pour chaque lettre une autre clé. Cette précaution est la base du *chiffrement de Vigenère*. Contrairement au chiffre César, qui, lui, est un *chiffrement monoalphabétique*, il s'agit ici d'un *chiffrement polyalphabétique*. Dans ce procédé, le nombre de rotations est répété surtout quand il s'agit de longs messages à chiffrer afin que la clé de déchiffrement ne soit pas trop longue. Malheureusement, quand le message est long et la clé est courte, le chiffrement est, en fin de compte, insuffisant et vulnérable aux attaques.

Sites web et mots clés

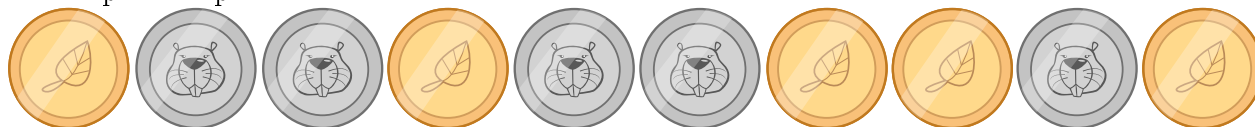
cryptographie, cryptage, chiffrement, déchiffrement, clé de déchiffrement, chiffrement monoalphabétique, chiffrement polyalphabétique, chiffre César, chiffrement de Vigenère

- https://fr.wikipedia.org/wiki/Chiffrement_par_décalage
- https://fr.wikipedia.org/wiki/Chiffre_de_Vigenère



22. Jeu des pièces

Christine possède dix pièces de monnaie qui ont soit une face dorée (🍂) soit une face argentée (🐻). Elle dispose ces pièces sur la table de la manière suivante :



Combien de fois doit-elle tourner un couple de deux pièces adjacentes pour qu'à la fin toutes les pièces montrent leur face dorée ? (Attention : il n'est possible de tourner que deux pièces de monnaie à la fois, ni plus, ni moins.)

- A) 1
- B) 2
- C) 4
- D) 6
- E) 8
- F) Ce n'est pas possible.



Solution

En effet, ce n'est pas possible.

Chaque fois qu'elle tourne deux pièces adjacentes, le nombre de pièces qui montrent la face argentée reste impair. On aura toujours soit deux faces argentées de plus, soit deux faces argentées de moins, soit le même nombre de faces argentées qu'avant : la parité des faces des pièces de monnaie reste la même.

La parité des faces argentées est au début impaire...et elle le restera jusqu'à la fin du jeu, car on n'arrivera jamais à tourner les pièces de sorte que le nombre de pièces avec une face argentée soit pair.

C'est de l'informatique !

On peut calculer les parités de manière rapide et simple. Les parités sont utiles pour vérifier si une transmission de données a été faite correctement (comme par exemple la lecture d'un code-barre) ou si un nombre a été correctement introduit (comme par exemple le numéro de compte dans le contexte des opérations bancaires en ligne). Quand il faut faire des calculs plus complexes, on peut même corriger directement quelques erreurs pour que les données soient transmises à nouveau.

Sites web et mots clés

parité, bit de parité

— https://fr.wikipedia.org/wiki/Somme_de_contrôle#Exemple:_bit_de_parité



23. Bar à jus de fruits

Sur leur route de vacances, quatre amis font une halte pour se rafraîchir dans un bar à jus de fruits. Chacun d'entre eux a ses propres préférences en ce qui concerne la saveur des jus. Celles-ci sont représentées dans le tableau ci-dessous. Plus il y a de cœurs, plus la personne en question préfère la saveur du jus indiquée. Anna préfère boire le jus marqué par trois cœurs au jus marqué par un seul cœur. Daniel, par contre, préfère boire le jus marqué par quatre cœurs au jus marqué par un cœur.

Anna				
Beat				
Christine				
Daniel				

Comme le bar à jus de fruits est très populaire, chacune des quatre saveurs ne peut être commandée qu'une seule fois.

Choisis pour chaque ami un jus de fruits afin que le nombre total des cœurs soit aussi grand que possible.



Solution

Le nombre maximum de cœurs que l'on peut atteindre est 14. Voici une des solutions possibles :

Anna				
Beat				
Christine				
Daniel				

Pour trouver la solution, on tiendra d'abord compte des préférences de Daniel. Il adore boire le jus de fruits , marqué par quatre cœurs, que les autres n'aiment pas autant (car ils ne l'ont marqué que par un seul cœur). Ensuite, si on clique sur le jus soit pour Beat soit pour Christine, on pourra choisir pour les amis qui n'ont pas encore de jus de fruits (soit Anna et Christine, soit Anna et Beat) le jus de fruits marqué par trois cœurs.

Trois des quatre amis préfèrent la saveur aux autres saveurs. Comme chaque saveur ne peut être commandée qu'une seule fois, deux amis sur trois doivent forcément se contenter de leur deuxième choix. Donc, pour arriver à un maximum de cœurs, le calcul se présentera comme suit : $3+3+4+4 = 14$.

D'ailleurs, toute autre solution demanderait à au moins un des amis de choisir le jus de fruits placé au troisième rang. Dans ce cas, le nombre maximum de cœurs sera 13 ($2 + 3 + 4 + 4 = 13$).

C'est de l'informatique !

La tâche demande d'optimiser le nombre de cœurs (et ainsi, d'optimiser la satisfaction des quatre amis). En programmation informatique ainsi qu'en mathématiques, *l'optimisation* représente un domaine de recherche important car elle est demandée dans de maintes situations en relation avec l'amélioration de l'efficacité d'un programme informatique. Dans le cas présent, un algorithme simple qui cherche à trouver toutes les solutions possibles dans un nombre fini d'étapes (ainsi que celles dont le calcul n'aboutira à aucun résultat), aura besoin de calculer 65000 différentes solutions. Grâce à des réflexions habiles, on arrive à réduire considérablement ce chiffre énorme (en effet, il n'y a que 24 solutions possibles à calculer). Malheureusement, ces réflexions ne sont pas si évidentes.

Notre tâche comporte en fait une forme particulière du *problème de couplage* : à chacune des quatre personnes, on doit attribuer une seule boisson et de chaque saveur il ne reste qu'un seul verre. Puis, il s'ajoute une autre condition à savoir que la satisfaction du groupe (le nombre maximum de cœurs



au total) doit être la plus haute possible. Nous trouvons de tels problèmes également au quotidien : il suffit de penser à la liste d'attente pour une transplantation d'organes. Dans ce cas, il faut tenir compte des *conditions* préalablement définies avant d'attribuer un organe à un patient comme par exemple le temps d'attente, le degré d'urgence établi par les spécialistes ou la compatibilité génétique entre le donneur et le receveur.

Sites web et mots clés

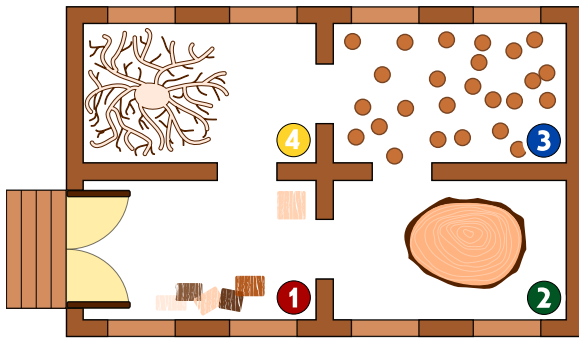
optimisation, couplage

- [https://fr.wikipedia.org/wiki/Optimisation_\(mathématiques\)](https://fr.wikipedia.org/wiki/Optimisation_(mathématiques))
- https://fr.wikipedia.org/wiki/Séparation_et_évaluation
- [https://fr.wikipedia.org/wiki/Couplage_\(théorie_des_graphes\)](https://fr.wikipedia.org/wiki/Couplage_(théorie_des_graphes))





24. Intrusion au musée



Le *Musée de la sculpture moderne sur bois* possède un système de sécurité très intéressant. Étant donné que les sculptures modernes sont passionnantes, les visiteurs se déplacent très lentement de salle en salle. À chaque minute, le système compte le nombre de personnes qui se trouvent dans une salle et il enregistre les valeurs obtenues dans un tableau. En outre, en se basant sur ces données, il contrôle si un intrus a tenté d'entrer au musée. Un intrus est une personne qui s'est introduite clandestinement dans

un lieu, ici, dans le musée, sans qu'elle soit entrée par la porte principale donc sans autorisation.

Dès que le système enregistre une infraction, c'est-à-dire dès qu'un intrus se trouve dans le musée, il sonne l'alarme.

Le tableau à gauche montre les enregistrements du système de sécurité de 10 :01 jusqu'à 10 :07.

L'image à droite montre le plan du musée avec les salles numérotées de 1 à 4.

heure	salle 1	salle 2	salle 3	salle 4
10 :01	2	0	0	0
10 :02	3	0	0	0
10 :03	2	1	0	0
10 :04	4	1	1	0
10 :05	2	2	3	0
10 :06	5	2	2	1
10 :07	4	1	2	2

À quelle minute le système de sécurité sonne-t-il l'alarme ?

- A) 10 :01
- B) 10 :02
- C) 10 :03
- D) 10 :04
- E) 10 :05
- F) 10 :06
- G) 10 :07



Solution

Le système de sécurité sonne l'alarme à 10 :05. À cette minute-là, le système compte dans la salle 3 deux personnes donc une personne de plus que dans la minute d'avant. À 10 :04, dans la salle d'à côté, à savoir dans la salle 2, par contre, il ne compte qu'une seule personne. Par conséquent, une personne s'est introduite dans la salle 3 sans être passée par la salle 2 ni par la salle 4, d'ailleurs, (car de 10 :04 à 10 :05, il n'y avait personne encore) et donc, sans avoir passé la porte d'entrée.

Pendant les autres heures, aucune infraction n'a été enregistrée :

- 10 :01, il y a 2 personnes dans la salle 1 ; on peut supposer qu'elles sont entrées par la porte d'entrée du musée.
- Entre 10 :01 et 10 :02, une autre personne entre dans la salle 1 ; rien n'indique qu'elle ne soit entrée au musée par la porte d'entrée.
- Entre 10 :02 et 10 :03, une personne passe de la salle 1 à la salle 2.
- Entre 10 :03 et 10 :04, une personne passe apparemment de la salle 2 à la salle 3 et une autre personne passe de la salle 1 à la salle 2. En même temps, il y a 3 autres personnes qui entrent dans le musée par la porte d'entrée.
- Entre 10 :05 et 10 :06, une personne passe de la salle 3 à la salle 4 et 3 autres personnes entrent dans le musée par la porte d'entrée.
- Entre 10 :06 et 10 :07, une personne passe de la salle 1 à la salle 4 et une autre personne passe de la salle 2 à la salle 1. En outre, il est fort probable qu'une autre personne quitte le musée parce qu'elle a terminé sa visite.

C'est de l'informatique !

Dans la vie quotidienne, on a recours à des systèmes de sécurité qui observent des personnes se trouvant dans des secteurs sensibles comme par exemple dans des aéroports. Des programmes informatiques exploitent les images de vidéo-surveillance en direct, notamment pour reconnaître une personne recherchée, ou ils analysent les données à d'autres fins précises. Aujourd'hui, ces programmes utilisent l'intelligence artificielle pour reconnaître des êtres humains, par exemple. Mais souvent, ils utilisent des règles simples comme dans la présente tâche pour détecter des problèmes de sécurité.

Sites web et mots clés

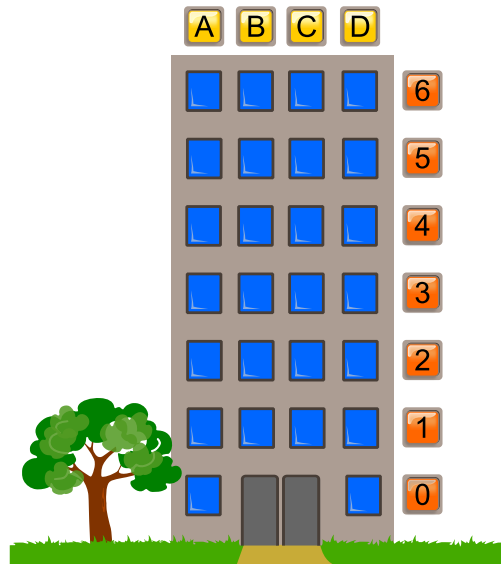
système de sécurité intelligent, règle, transition d'état

- https://fr.wikipedia.org/wiki/Système_de_détection_d'intrusion

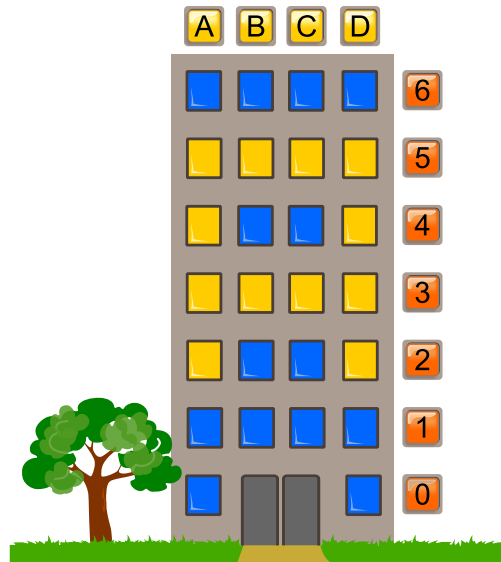


25. Des jeux de lumière

Dans la nouvelle tour de la ville, il y a un contrôle centralisé qui règle l'allumage des lumières. La tour comporte 26 fenêtres à travers lesquelles on peut voir si, à l'intérieur, les lumières sont allumées ou éteintes. Malheureusement, il n'est pas possible de régler les lumières séparément : soit on éclaire un étage entier, soit on éclaire une colonne de fenêtres entière.



Sur quels étages (à indiquer par leurs numéros) ou pour quelles colonnes (à indiquer par leurs lettres) faut-il allumer ou éteindre pour que les fenêtres s'allument comme montré ici ?





Solution

On peut résoudre ce problème de manière simple : on allume d'abord les fenêtres des étages no 3 et no 5, ensuite, on éteint les fenêtres des colonnes A et D. Finalement, on doit éteindre les lumières des étages no 6, 1 et 0. Il existe évidemment beaucoup d'autres solutions possibles, mais toutes les solutions possibles ont recours à cette même séquence d'opérations.

C'est de l'informatique !

On peut comparer les interrupteurs de cette tâche avec des instructions de n'importe quel système ou de n'importe quel appareil informatique. Même les programmes informatiques beaucoup plus complexes peuvent être compris comme une séquence d'instructions simples. Une fenêtre correspondra donc à une espace mémoire qui, elle, peut avoir la valeur 0 (lumière éteinte) et 1 (lumière allumée). Les ordinateurs modernes permettent l'exécution d'instructions composites qui gèrent parallèlement plusieurs espaces mémoire. Dans notre tâche, ces instructions-là correspondraient aux interrupteurs des fenêtres d'un étage ou d'une colonne. En général, les ordinateurs actuels manipulent simultanément plusieurs millions d'espaces mémoire et ceci, à l'intérieur du processeur seulement... la mémoire vive et le disque dur sont encore plus puissants car ils sont capables de manipuler plusieurs milliards et même milliards d'espaces mémoire !

C'est pourquoi il est important que les instructions composites soient clairement définies à savoir que l'on détermine le moment où elles seront exécutées (*pre-condition*) et ce qui suivra à leur exécution (*post-condition*). Dans notre exemple, les conditions suivantes sont valables pour les lumières d'un étage : si une seule lumière est éteinte (*pre-condition*), toutes les autres lumières de cet étage sont allumées au moment où l'interrupteur est actionné (*post-condition*). Sinon (dans le cas où toutes les lumières sont allumées, *pre-condition*), toutes les lumières de l'étage seront éteintes (*post-condition*).

Sites web et mots clés

langage assembleur (programmation informatique), séquences, opérations en codage binaire

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



26. Substitutions

M. Müller est tombé brusquement malade. Dans l'entreprise où il travaille, M. Maier doit le remplacer et accomplir toutes les tâches dont M. Müller était responsable. Heureusement, M. Müller se rétablit plus vite que prévu et retourne au travail deux semaines plus tard. Comme M. Maier a très bien travaillé, les deux collègues décident qu'à partir de maintenant, M. Maier continuera à accomplir les tâches de M. Müller et que M. Müller accomplira les tâches de M. Maier. Par conséquent, la documentation du projet en cours doit être changée comme suit : le nom de M. Müller doit être substitué au nom de M. Maier et vice versa. Dans la documentation, il est possible de substituer chaque texte à un autre.

Laquelle des démarches suivantes est valable si l'on suppose que le texte ne comporte aucun symbole «#» ?

- A) Je remplace d'abord tous les «Müller» par «Maier» et puis tous les «Maier» par «Müller».
- B) Je remplace d'abord tous les «Maier» par «Müller» et puis tous les «Müller» par «Maier».
- C) Je remplace d'abord tous les «Müller» par le symbole «#», ensuite le symbole «#» par «Maier» et finalement les «Maier» par «Müller».
- D) Je remplace d'abord tous les «Müller» par le symbole «#», ensuite tous les «Maier» par «Müller» et finalement les «#» par «Maier».



Solution

La réponse correcte est D) Je remplace d'abord tous les «Müller» par le symbole «#», ensuite tous les «Maier» par «Müller» et finalement les «#» par «Maier».

- A) Dans ce cas-là, il ne restera plus que le nom «Müller» et tous les «Maier» auront été supprimés car suite à la première substitution, le texte ne comportera que le nom «Maier» qui, lui, sera remplacé par «Müller».
- B) Dans ce cas-là, il ne restera que le nom «Maier» et tous les «Müller» auront été supprimés car suite à la première substitution, le texte ne comportera que le nom «Müller» qui, lui, sera remplacé par «Maier».
- C) Dans ce cas-là, il ne restera que le nom «Müller» et tous les «Maier» auront été supprimés car après avoir remplacé tous les «Müller» par le symbole «#», celui-ci est immédiatement remplacé par «Maier» et finalement on remplace l'ensemble des «Maier» par «Müller».
- D) C'est la seule procédure qui aboutira au résultat voulu : les «Müller» seront temporairement remplacés par le symbole «#» et restent ainsi enregistrés tandis que les «Maier» seront tous remplacés par «Müller».

C'est de l'informatique !

Bien que la procédure de substitution soit très simple, elle est très importante en informatique. Grâce à de telles substitutions, les informaticiens peuvent effectuer des opérations complexes. En informatique théorique, par exemple, les grammaires formelles (à la base des langages de programmation) sont définies comme une liste de règles de remplacement.

Notre exercice insiste sur le fait que l'on a souvent besoin d'un élément temporaire pour pouvoir effectuer l'échange de deux valeurs : ce concept est à la base de l'échange de variables (swap).

Sites web et mots clés






traitement de texte, suivre des séquences d'instructions, échange de variables (swap)

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



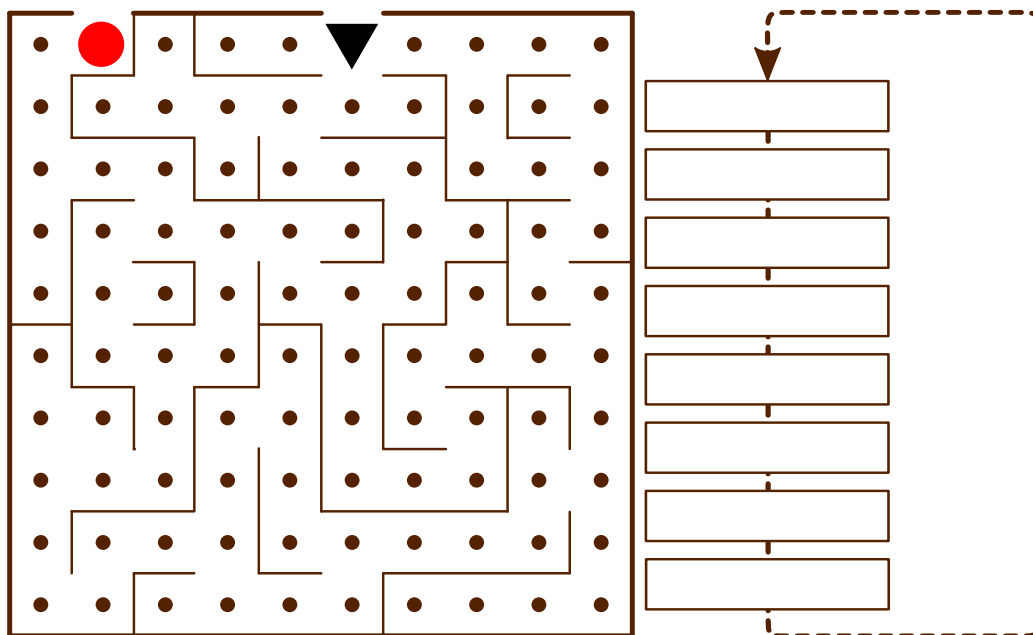
27. Sors du labyrinthe !

Benjamin aimerait bien traverser un labyrinthe. Comme il n'est pas habitué à ce genre d'exercice, il te demande de l'aider. Il commence au point de départ qui est un triangle noir et il envisage d'atteindre la sortie marquée par un cercle rouge. Cependant, Benjamin ne peut mémoriser qu'une série de huit instructions composée des instructions suivantes :

		Fais un pas en avant, ensuite, tourne à gauche.
		Fais un pas en avant, ensuite, tourne à droite.
		Fais un pas en avant.

Même si Benjamin ne peut mémoriser qu'une série de huit instructions au maximum, il peut effectuer les différentes instructions de manière répétitive jusqu'à ce qu'il sorte du labyrinthe.

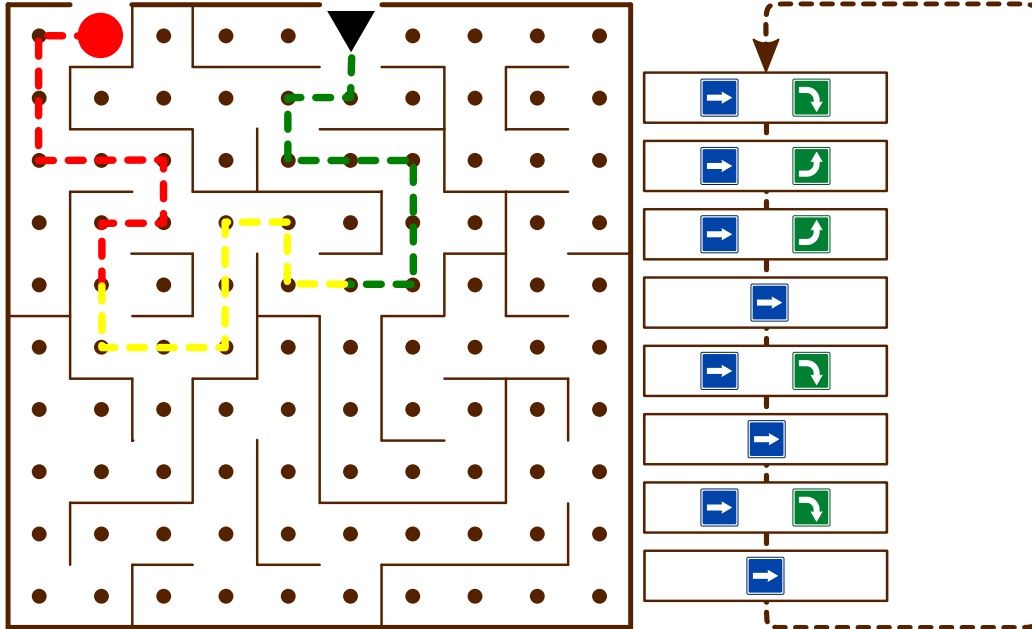
Au début, Benjamin se trouve sur le point de départ, le regard tourné vers le bas. Insère les instructions dans le bloc d'instructions ci-dessous pour que Benjamin puisse atteindre la sortie.





Solution

La série d'instructions suivante mènera Benjamin jusqu'à la sortie à condition qu'elle soit effectuée trois fois :



C'est de l'informatique !

Au fond, Benjamin exécute un programme informatique. Ce programme comprend une série d'instructions (une *séquence* ou un *bloc d'instructions*). Une *structure de contrôle* comme la *boucle* (en anglais, *loop*) dans notre programme permet de réutiliser une série d'instructions jusqu'à ce qu'un résultat particulier soit obtenu ou qu'une condition prédéterminée soit remplie. Une telle structure permet d'ailleurs de limiter le nombre d'instructions et, finalement, d'économiser beaucoup de temps. De plus, on arrive à repérer et à corriger plus facilement les erreurs éventuelles d'un programme.

Sites web et mots clés

série d'instructions, séquence, bloc d'instructions, boucle, algorithme

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

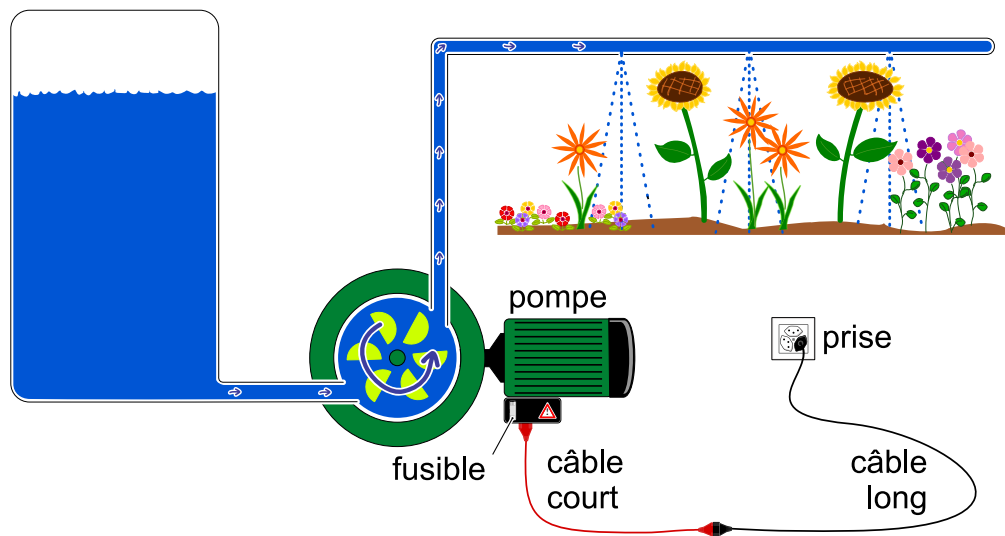


28. Système d'arrosage

M. Lejardinier est fier propriétaire d'un parterre fleuri ainsi que d'un potager. Pour que toutes ses plantes prospèrent, il a construit deux systèmes d'arrosage identiques. Voilà une image de l'un des deux systèmes ci-dessous.

Ce système puise son énergie d'un réseau d'alimentation électrique et se compose :

- d'un câble long ;
- d'un câble court ;
- d'une pompe . . .
- . . .pourvue d'un fusible (la pompe s'arrête quand le fusible saute).



Un jour, M. Lejardinier remarque que son système d'arrosage du parterre fleuri ne fonctionne plus. Après avoir vérifié le réservoir ainsi que toutes les conduites d'eau, qui, elles, sont en bon état, il se rend compte qu'il doit aller à la recherche d'une partie spécifique du système qui est cassée. Comme il veut absolument trouver une solution à son problème, il envisage d'échanger toutes les parties du système d'arrosage du parterre fleuri avec les parties de celui du potager.

Coche d'une croix toutes les affirmations correctes que tu trouveras ci-dessous.

- A) Pour trouver la ou les causes du problème, il faut qu'il commence par la pompe puisque c'est la partie la plus importante d'un système d'arrosage.
- B) Il est possible de remplacer autant de parties que nécessaire du système défectueux avec les parties du système d'arrosage qui fonctionne parfaitement pourvu qu'on les échange progressivement, c'est-à-dire étape par étape. Si le système d'arrosage redémarre après l'échange de la dernière partie, ce sera celle-ci qui est défectueuse.
- C) On peut vérifier d'abord si la prise est coupée en y branchant un autre appareil électronique. Si cet appareil fonctionne, les causes du problème se trouveront probablement ailleurs.
- D) Il vaudrait mieux acheter des parties neuves que de réutiliser des parties du système d'arrosage du potager qui, elles, sont déjà usées.
- E) Pour être plus efficace et pour progresser plus rapidement, il est recommandé d'échanger deux parties en même temps à chaque étape.



Solution

Les réponses B) et C) sont correctes.

Réponse A) : l'importance attribuée à une partie d'un système ne nous renseigne nullement sur la probabilité d'une erreur.

Réponse D) : des parties neuves, elles aussi, peuvent être défectueuse. Les parties provenant du système d'arrosage du potager, par contre, sont apparemment en bon état puisque ce système fonctionne actuellement bien.

Réponse E) : en effet, s'il ne s'agissait que de réparer le système le plus vite possible, cette variante permettrait d'avancer plus vite.

C'est de l'informatique !

Comme les installations complexes peuvent parfois être sensibles aux erreurs qui provoquent des dysfonctionnements, la vérification et l'élimination des erreurs est une tâche très importante. En informatique, on parle du *débogage*. Repérer et éliminer une seule erreur, appelé *bogue*, constitue un véritable défi. Repérer deux ou plusieurs bogues est particulièrement difficile. Pour être plus efficaces et pour être sûr que l'on ne cherche qu'un seul bogue, les développeurs d'un logiciel qui analyse les bogues d'un programme informatique préfèrent effectuer des tests après chaque étape de modification.

Sites web et mots clés

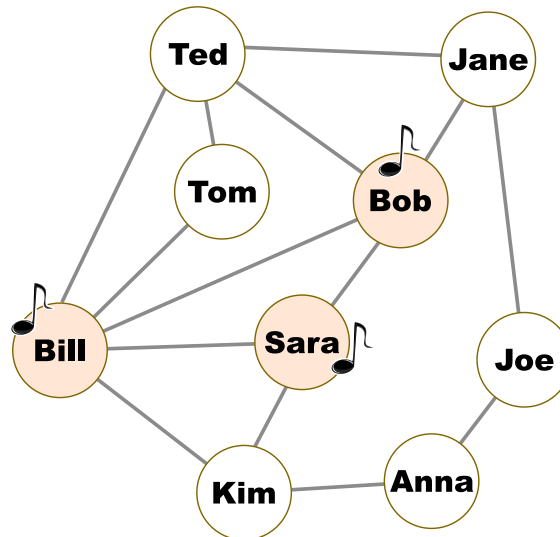
vérification, débogage

- <https://en.wikipedia.org/wiki/Systemantics>
- https://en.wikiquote.org/wiki/John_Gall
- <https://en.wikipedia.org/wiki/Debugging>



29. Une nouvelle chanson

Dans le diagramme suivant, deux personnes sont amis seulement si leurs prénoms sont reliés par une ligne. Lundi dernier, une mégastar a sorti son tube le plus récent. Le jour même, Bill, Bob et Sara ont acquis la nouvelle chanson. Leurs noms sont marqués avec une note de musique.



À partir de mardi, il se passe la chose suivante : toutes les personnes du groupe achètent la chanson qui a déjà été achetée la veille par au moins la moitié de leurs amis. Ainsi, Tom, par exemple, a acheté la chanson mardi tandis que Jane ne l'a pas encore achetée.

Quel est le jour précis où toutes les personnes du groupe auront acquis cette chanson au plus tôt ?

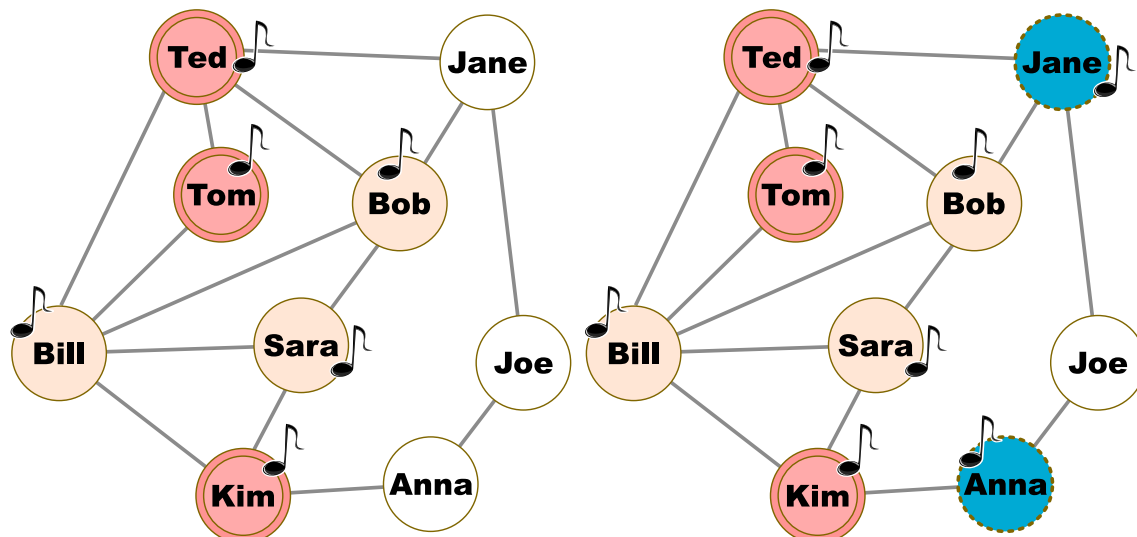
- A) Mercredi
- B) Jeudi
- C) Vendredi
- D) Samedi



Solution

La réponse B) est correcte.

Tom, Ted et Kim achètent la chanson mardi, Anna et Jane l'achètent mercredi :



Finalement, Joe l'achète jeudi. Jeudi est donc le jour où toutes les personnes du groupe seront en possession de la chanson.

C'est de l'informatique !

Le nombre des utilisateurs des réseaux sociaux s'élève souvent à bien plus qu'un milliard. C'est la raison pour laquelle ces plate-formes sont également intéressantes pour des sociétés de marketing : ils s'en servent assez fréquemment pour diffuser des informations concernant de nouveaux produits. Notre tâche se base en fait sur ce processus de diffusion.

Pour simuler un tel processus, nous avons recours à un modèle à seuil. Le diagramme de notre tâche est un graphe qui comprend des nœuds (représentés par les prénoms des personnes) et des arêtes (représentées par les lignes entre les nœuds ; dans le contexte des réseaux sociaux, ces lignes représentent les amitiés entre les utilisateurs). Les nœuds qui sont reliés par les arêtes sont appelés «voisins». Le seuil d'infection « q » est un chiffre entre 0 et 1. Dans notre tâche, q est égal à 0,5. Une personne montre donc un nouveau comportement B (elle achète, par exemple, une nouvelle chanson) si au moins une part égale à q de ses voisins du graphe montre le même comportement.

Sites web et mots clés

réseau social, graphe

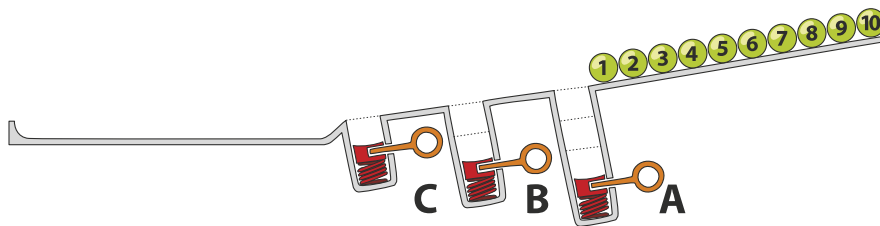
— https://fr.wikipedia.org/wiki/Étude_du_petit_monde



30. Jeu de billes

Sur une rampe, il y a 10 billes numérotées. Le long de la rampe, il y a trois trous A, B et C : le trou A peut contenir trois billes au maximum, le trou B deux billes et le trou C une seule bille au maximum. Quand les billes roulent sur la rampe, elles tombent successivement dans les trous jusqu'à ce qu'elles les remplissent (les billes 1, 2 et 3 tombent dans le trou A, les billes 4 et 5 tombent dans le trou B et la bille 6 tombe dans le trou C). Les autres billes passent par-dessus et continuent leur chemin jusqu'à la fin de la rampe.

Quand toutes les billes ont parcouru la rampe, les ressorts, placés dans les trous A à C, éjectent les billes qu'ils contenaient : d'abord, les trois billes du trou A, ensuite, celles du trou B et finalement, celle du trou C. Les billes sont ainsi poussées sur la rampe. On attend que toutes les autres billes aient passé avant qu'un ressort ne soit relâché.



Dans quel ordre les billes de la séquence 1 à 10 seront-elles alignées à la fin ?

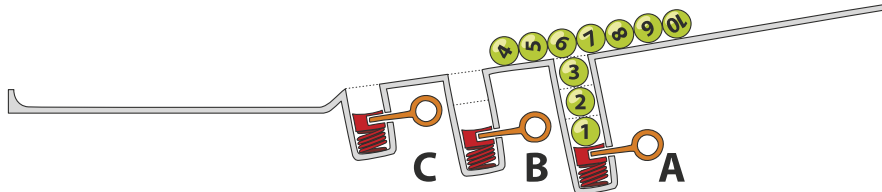
- A) B) C) D)



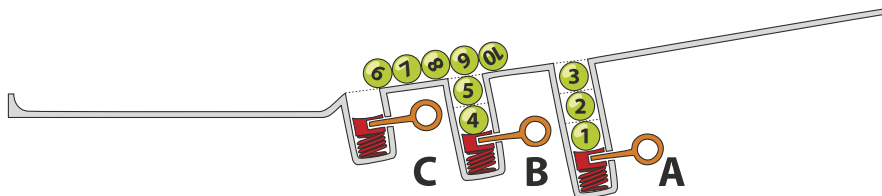
Solution

La réponse correcte est D) 7 8 9 10 3 2 1 5 4 6.

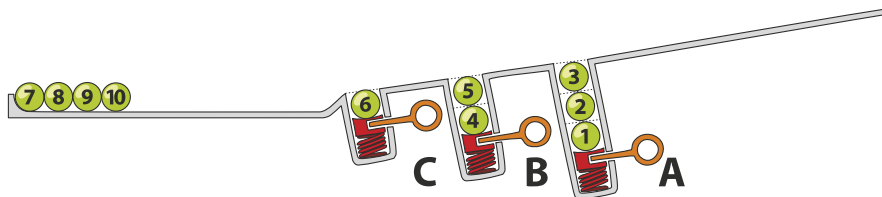
Les billes 1, 2 et 3 tombent dans le trou A, les billes 4 à 10 passent le trou A (qui, lui, contient les billes 1 à 3).



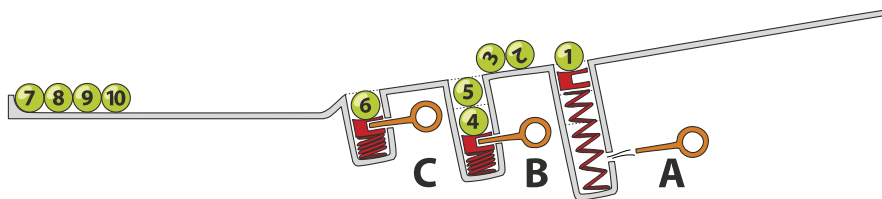
Ensuite, les billes 4 et 5 tombent dans le trou B et les billes 6 à 10 passent le trou B (qui, lui, contient les billes 4 à 5).



Finalement, la bille 6 tombe dans le trou C et les billes 7 à 10 passent le trou C (qui, lui, contient la bille 6). Ces-dernières atteignent la fin de la rampe dans l'ordre de 7 à 10.

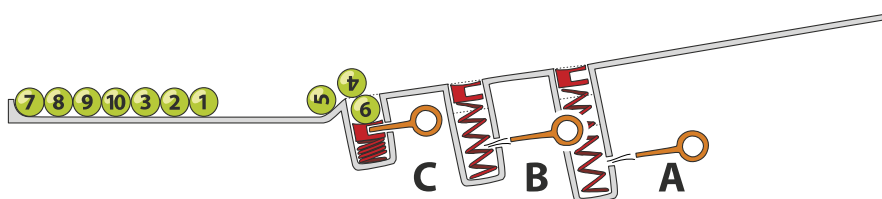


Quand les billes 7 à 10 ont passé les trous, le ressort du trou A libère les billes 3, 2, 1. Celles-ci se retrouvent donc de nouveau sur la rampe et roulent en direction de la fin de la rampe.



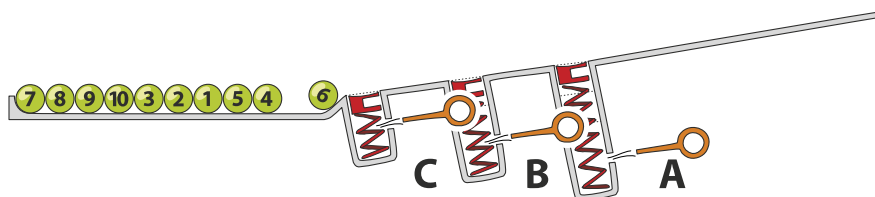
L'ordre de la séquence est donc 7, 8, 9, 10, 3, 2, 1.

Ensuite, le ressort du trou B libère les billes 5 et 4. Celles-ci roulent également en direction de la fin de la rampe. Finalement, le ressort du trou C libère la bille 6 qui s'ajoute aux billes à la fin de la rampe.





Toutes les billes arrivent donc à la fin de la rampe dans l'ordre suivant : 7, 8, 9, 10, 3, 2, 1, 5, 4, 6.



C'est de l'informatique !

Les trous de la rampe rappellent la structure de la pile (en anglais, *stack*). En informatique, une pile permet d'enregistrer des données pour les réutiliser ensuite selon le principe de *Last-In First-Out (LIFO)* (en français «dernier arrivé, premier sorti») : ainsi, la dernière bille qui est tombée dans le trou sera libérée la première. Aussi simple que ce principe puisse paraître, il est très utile dans bon nombre de situations. On peut par exemple vérifier si les parenthèses d'une expression arithmétique sont équilibrées (dans l'expression $((1 + 2) \cdot 3)$, les parenthèses sont équilibrées tandis que dans l'expression $((4 + 5) \cdot (6 - 7))$, elles ne le sont pas), en procédant comme suit : chaque parenthèse ouvrante est ajoutée à la pile (à l'aide de l'opération appelée *push*) et quand une parenthèse fermante apparaît, la parenthèse ouvrante est enlevée de la pile (à l'aide de l'opération appelée *pop*). Si on ne trouve plus de parenthèse ouvrante à enlever de la pile bien que l'on en ait encore besoin, ou si à la fin de l'expression il en reste encore dans la pile, nous pouvons être sûrs qu'il y a une erreur. Contrairement à cela, si à la fin de l'expression la pile ne contient plus de parenthèse, cela signifie que les parenthèses sont équilibrées et que donc l'expression est correcte.

Sites web et mots clés

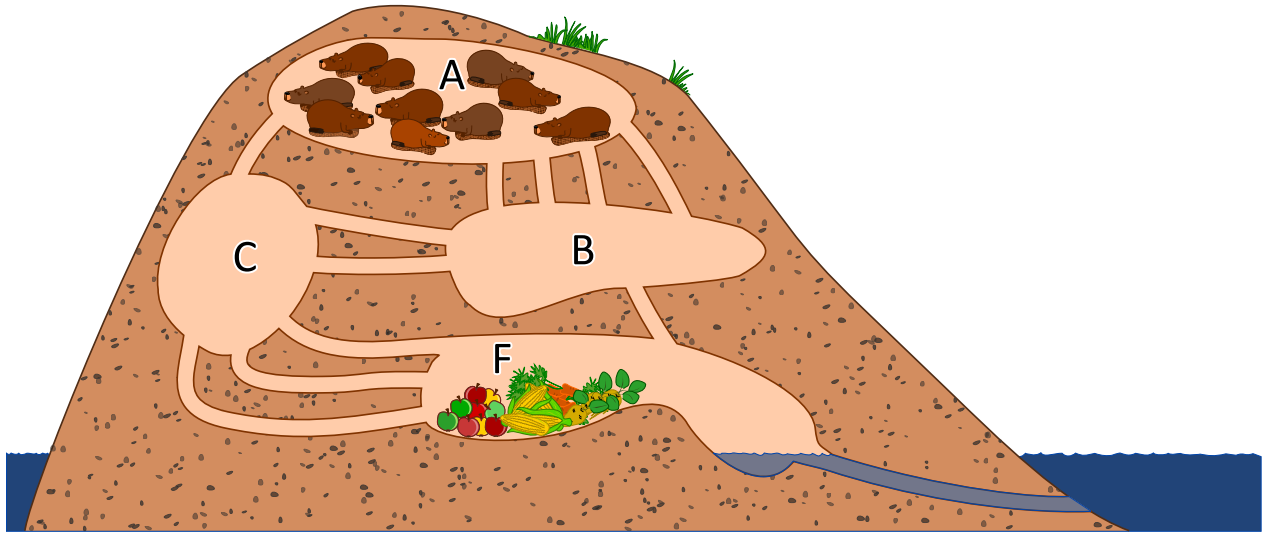
Stack (en français pile), LIFO

- https://fr.wikipedia.org/wiki/Last_in,_first_out
- [https://fr.wikipedia.org/wiki/Pile_\(informatique\)](https://fr.wikipedia.org/wiki/Pile_(informatique))





31. À table, mais vite !



10 castors se trouvent dans la chambre A. Ils aimeraient bien se déplacer le plus vite possible dans la chambre F pour aller manger leur dîner. Chaque castor a besoin d'une minute pour parcourir un couloir qui relie deux chambres et il n'est pas possible que deux castors parcourent un couloir en même temps. Dans les chambres A, B, C, F, il y a cependant assez de place pour tous les castors et la traversée d'une chambre ne prend pas de temps.

Après combien de minutes le groupe des 10 castors se retrouvera-t-il dans la chambre F? Indique le temps le plus court possible.



Solution

Le groupe des 10 castors arrivera dans la chambre F après 4 minutes seulement.

Dans leur terrier, il y a deux chemins qui sont les plus courts pour passer de la chambre A à la chambre F. Chacun des deux chemins peut être parcouru par un castor en deux minutes :

- A → B → F
- A → C → F

Après 2 minutes, il y aura donc 2 castors dans la chambre F et après 3 minutes, encore deux autres castors.

Le chemin A → B → C → F permettra à deux castors de passer de la chambre A à F, ce parcours prenant 3 minutes. Ainsi, après 3 minutes, il y aura 6 castors qui se retrouveront dans la chambre F (4 d'entre eux choisiront les deux chemins les plus courts et 2 d'entre eux le chemin plus long). À la 4e minute, tous les 10 castors se retrouveront dans la chambre F. Le tableau suivant précise les différents déplacements :

déplacements / situation	Nombre de castors dans les chambres (suite à un déplacement)			
	A	B	C	F
Situation de départ	10	0	0	0
<i>3 castors se déplacent de A à B (moins que ce qui serait possible)</i> <i>1 castor se déplace de A à C</i>				
Situation après 1 minute	6	3	1	0
<i>3 castors se déplacent de A à B (moins que ce qui serait possible)</i> <i>1 castor se déplace de B à F</i> <i>2 castors se déplacent de B à C</i> <i>1 castor se déplace de C à F</i> <i>1 castor se déplace de A à C</i>				
Situation après 2 minutes	2	3	3	2
<i>1 castor se déplace de A à B (le chemin le plus court)</i> <i>1 castor se déplace de B à F</i> <i>2 castors se déplacent de B à C</i> <i>1 castor se déplace de A à C (le chemin le plus court)</i> <i>3 castors se déplacent de C à F</i>				
Situation après 3 minutes	0	1	3	6
<i>1 castor se déplace de B à F</i> <i>3 castors se déplacent de C à F</i>				
Situation après 4 minutes	0	0	0	10

En effet, il existe plusieurs solutions qui démontrent comment les 10 castors peuvent parcourir les couloirs afin d'arriver en 4 minutes dans la chambre F. Mais dans la présente solution, aucun des castors n'est contraint à attendre dans une chambre intermédiaire jusqu'à ce qu'il puisse continuer son chemin.

C'est de l'informatique !

Le réseau des couloirs du terrier de castor peut être compris comme un *réseau de flot* en informatique. Le nombre des couloirs entre deux chambres détermine le nombre de castors qui peuvent le parcourir



en une minute. On parle de *capacité* de connexion entre les deux chambres. Celle-ci limite le *flot* (ou flux) maximal.

En théorie de graphes, un réseau de flot est un *graphe orienté* dont chaque *arête* (dans notre tâche, il s'agit d'un couloir) possède une capacité maximale. Un flot qui parcourt les arêtes est donc limité par la capacité de ces-dernières. À l'aide des réseaux de flot, il est possible de simuler un réseau informatique ou un réseau de transport pour dévoiler, par exemple, les zones sensibles qui pourraient provoquer des problèmes routiers tels que des embouteillages.

Dans le domaine des réseaux de flot, il est particulièrement intéressant d'analyser le flot maximum entre deux points appelés *noeuds*. Dans notre tâche, il y a au maximum 4 castors par minute qui puissent se déplacer de la chambre A à F sans devoir attendre dans une des chambres intermédiaires. Il existe un algorithme appelé *algorithme de Ford-Fulkerson* à l'aide duquel on peut calculer un tel flot maximum.

Sites web et mots clés

réseau de flot, capacité, graphe, graphe orienté, arête, flot, flot dans le réseau, noeud, algorithme de Ford-Fulkerson

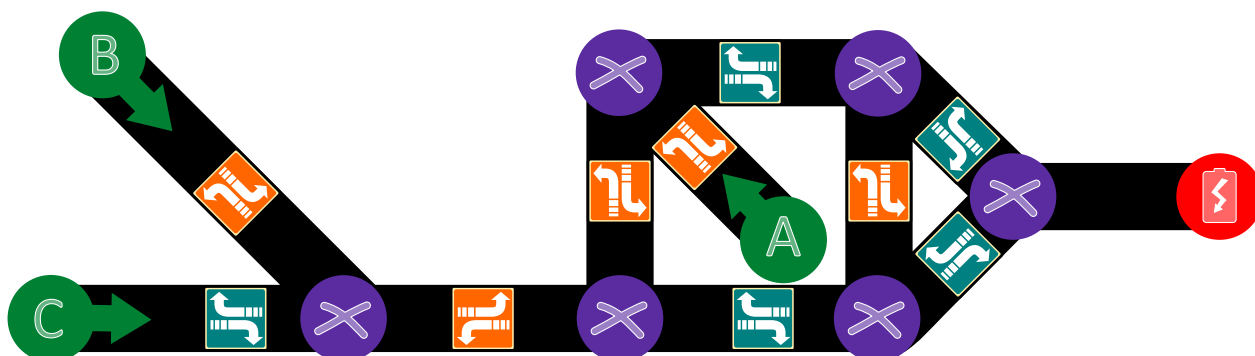
- https://fr.wikipedia.org/wiki/Réseau_de_flot
- https://fr.wikipedia.org/wiki/Graphe_orienté
- https://fr.wikipedia.org/wiki/Algorithme_de_Ford-Fulkerson





Solution

La solution correcte est la suivante :



Pour les deux lignes qui partent des points B et C, nous allons nous assurer que l'Arabot puisse continuer vers la droite. C'est pourquoi Jonas doit insérer l'instruction pour la ligne qui part du point B et l'instruction pour la ligne qui part du point C.

Dans la partie droite du parcours, pour que l'Arabot arrive à la station de recharge (), la première ligne verticale doit comporter l'instruction sinon, le robot de Jonas arrivera au point A où il s'éteindra. La ligne horizontale tout en haut du parcours, par contre, doit comporter obligatoirement l'instruction sinon l'Arabot fera fausse route. De la même manière, la ligne verticale à droite doit comporter l'instruction . Finalement, Jonas doit insérer l'instruction dans la ligne horizontale tout en bas du parcours pour que l'Arabot tourne à droite et continue dans la direction de la station de recharge ().

C'est de l'informatique !

L'objectif de cette tâche est de codifier les divers chemins qui mènent à un certain but (de «A» à , de «B» à ou de «C» à) avec une structure appropriée (dans notre cas, il s'agit d'un graphe). En informatique, une telle structure est appelée une *structure de données*. Quand l'Arabot suit un certain parcours (par exemple, de «A» à , il doit lire et exécuter les instructions préalablement déterminées : «*Quelle direction dois-je prendre quand j'arrive au prochain carrefour ? Si l'instruction est claire, je sais quelle direction je dois prendre.*» Au niveau du matériel informatique, l'ordinateur fonctionne de la même manière : il lit les instructions et il les exécute.

Cette tâche nous révèle qu'il existe de nombreuses questions intéressantes du point de vue mathématique ou informatique, pourtant elle nous montre aussi à quel point il est difficile de déterminer les instructions de manière évidente. Dans ce contexte, il y a beaucoup de questions ouvertes. Les informaticiens appartenant au domaine de la recherche des *algorithmes* et de la *théorie de complexité* s'intéressent à de tels problèmes. Tout comme les scientifiques des domaines de la *biologie* et de la *médecine computationnelle*.

Sites web et mots clés

graphe orienté, théorie de complexité, biologie computationnelle, médecine computationnelle

- [https://fr.wikipedia.org/wiki/Théorie_de_la_complexité_\(informatique_théorique\)](https://fr.wikipedia.org/wiki/Théorie_de_la_complexité_(informatique_théorique))
- https://en.wikipedia.org/wiki/Computational_biology
- https://en.wikipedia.org/wiki/In_silico_medicine



33. Les piles de cure-dents à diviser

Hélène et Bob jouent à un jeu qui se base sur des piles de cure-dents. Le jeu démarre avec deux piles. Chaque joueur, à son tour, ...

1. ...doit mettre une des deux piles de côté...
2. ...et diviser la pile restante en deux.

Un joueur gagne quand il laisse deux piles, chacune comportant un seul cure-dent. C'est au tour d'Hélène.

Pour commencer, Hélène choisit la pile avec 24 cure-dents qu'elle doit diviser en deux. Comment doit-elle la diviser pour qu'elle gagne le jeu ?

- A) en 11 et 13
- B) en 12 et 12
- C) en 7 et 17
- D) en 8 et 16



Solution

Pour gagner le jeu, Hélène a deux possibilités : soit elle divise la pile en 11 et 13 soit en 7 et 17.

Pour qu'elle puisse gagner le jeu, il est nécessaire qu'elle divise la pile en deux piles impaires. Si elle divise la pile en deux piles paires, elle concédera la victoire à Bob. . . D'ailleurs, selon les règles mathématiques, il n'est pas possible de diviser une pile de 24 cure-dents en une pile paire et en une autre pile impaire.

Si Hélène veut gagner le jeu, pourquoi doit-elle impérativement utiliser la stratégie décrite ci-dessus ? La réponse est simple : si un joueur laisse à l'autre joueur deux piles impaires, ce dernier ne peut que laisser à son tour une pile paire et une pile impaire. Ensuite, son adversaire mettra de côté la pile impaire et divisera la pile paire en deux piles impaires. Comme mentionné un peu plus haut, le jeu se termine quand il ne reste que deux piles à un cure-dent. . . donc, deux piles impaires. Ce qui revient à dire que le gagnant du jeu sera toujours celui qui laissera à son adversaire une pile impaire de cure-dents.

C'est de l'informatique !

On peut facilement trouver des stratégies gagnantes pour de tels jeux si on arrive à trouver une invariante (c'est-à-dire une qualité qui ne changera pas pendant le cours du jeu). Celle-ci mènera dans tous les cas à la victoire. Dans notre tâche, pour gagner le jeu, l'invariante est qu'il faut toujours laisser à son adversaire une pile impaire de cure-dents.

Dans le présent jeu, à part d'utiliser la bonne stratégie, il est également important d'éviter que l'adversaire se serve d'abord de ladite stratégie : ainsi, il est non seulement important de tenir compte de la situation de départ (soit le nombre des cure-dents), mais aussi de savoir lequel des deux joueurs commencera le jeu car le début du jeu a un impact sur le résultat de la partie si les deux joueurs se comportent de manière optimale.

Les informaticiens s'occupent souvent des jeux où rien n'est laissé au hasard et où tout dépend de la bonne stratégie. Jouer à ce type de jeu peut s'avérer très instructif : dans ce contexte, il n'est pas important qu'il s'agisse de jeux simples comme dans notre tâche dont la stratégie peut être calculée par un ordinateur simple en quelques secondes seulement ou qu'il s'agisse, au contraire, de jeux plus complexes comme un jeu d'échecs ou le jeu de Go pour lesquels les ordinateurs les plus puissants ont besoin de plusieurs années parfois pour calculer le meilleur mouvement. Grâce à ces *jeux de stratégie* dits *combinatoires abstraits*, nous nous exerçons dans notre capacité à prendre la juste décision dans des situations les plus complexes. . . et ceci non seulement dans un jeu vidéo mais aussi, par exemple, dans des applications de l'intelligence artificielle.

Sites web et mots clés

jeux de stratégie, arbre de décision

- https://fr.wikipedia.org/wiki/Jeu_de_stratégie_combinatoire_abstrait
- https://fr.wikipedia.org/wiki/Théorie_des_jeux_combinatoires



34. Calculer la distance entre les mots

Pour calculer la distance entre deux mots, il est recommandé d'effectuer les opérations suivantes :

- ajouter une lettre à un endroit donné du mot
- supprimer une lettre à un endroit donné du mot
- substituer une lettre par une autre à un endroit donné du mot

La distance entre deux mots est égale au nombre minimal de ces opérations élémentaires qui sont nécessaires pour transformer le premier mot en le second.

Ainsi, la distance entre les deux mots «plier» et «ramer» est 4, comme nous pouvons le voir dans l'exemple suivant :

1. plier → prier (substituer la lettre «l» par la lettre «r»)
2. prier → primer (ajouter la lettre «m»)
3. primer → rimer (supprimer la lettre «p»)
4. rimer → ramer (substituer la lettre «i» par la lettre «a»)

Quelle est la distance entre les deux mots «Emil» et «Erich» ?



Solution

La distance entre les deux mots «Emil» et «Erich» est 3. Nous pouvons, par exemple, effectuer les opérations suivantes :

Emil → Eril → Eric → Erich

Bien qu'il existe plusieurs solutions, le nombre d'opérations ne sera jamais au-dessous de 3 : le prénom Erich comporte une lettre de plus que le prénom Emil (une opération) et le prénom Erich ne comporte ni la lettre «m» ni la lettre «l» (deux opérations).

C'est de l'informatique !

On appelle la distance entre les mots la *distance de Levenshtein*. Elle a été découverte par le Russe Vladimir Levenshtein en 1965. Dans le contexte de l'assistance orthographique, la distance de Levenshtein propose, par exemple, automatiquement la correction d'une erreur d'orthographe : si la distance entre le mot erroné et le mot proposé par l'assistance orthographique est petite, il est fort probable que l'on n'a fait qu'une erreur de frappe. La méthode du calcul des distances est utilisée dans de différents domaines : par exemple dans la recherche génétique quand il s'agit de calculer la ressemblance des brins ADN, dans le domaine des images ou encore dans le domaine de la traduction automatique des textes.

Il est possible de calculer la distance de Levenshtein à l'aide d'un programme informatique en essayant toutes les substitutions possibles. Dans ce cas, pour éviter que le programme ne calcule des mots inutilement longs, on met en place des délimitations précises à l'intérieur desquelles il peut effectuer ses opérations.

Sites web et mots clés

distance de Levenshtein, distance d'édition

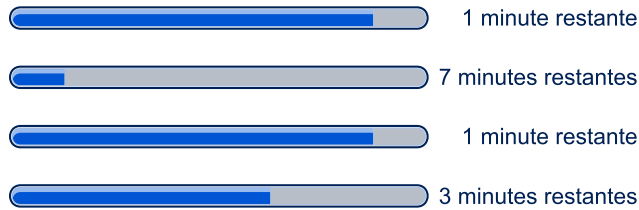
- https://fr.wikipedia.org/wiki/Distance_de_Levenshtein
- https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance



35. Des téléchargements en parallèle

Quand on télécharge plusieurs fichiers volumineux en même temps, la capacité de connexion est divisée par le nombre de fichiers à télécharger. Par exemple, quand on télécharge 10 fichiers en même temps, chaque fichier ne disposera que d'un dixième de la capacité de connexion.

Lorsqu'un utilisateur télécharge 4 fichiers en même temps, le temps restant est calculé en fonction de la vitesse de transmission actuelle :



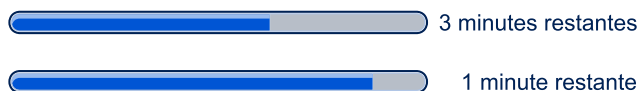
Combien de minutes l'utilisateur doit-il attendre jusqu'à ce que les 4 fichiers soient téléchargés ?



Solution

Le téléchargement des 4 fichiers prendra au total 3 minutes.

Après une minute, deux des quatre fichiers seront entièrement téléchargés. Selon les barres de progression, le temps nécessaire au téléchargement des deux fichiers manquants s'élève encore à 6 et à 2 minutes. Cependant, la vitesse de téléchargement sera doublée dès cet instant car ces deux fichiers disposeront maintenant d'une capacité de connexion double. Il leur faut donc 3 et 1 minutes pour compléter leur téléchargement :



Deux minutes plus tard, le troisième fichier sera entièrement téléchargé. Selon la barre de progression, l'utilisateur devra attendre encore 2 minutes jusqu'à ce que le dernier fichier soit téléchargé. Mais comme la vitesse sera doublée une fois encore, pour la même raison que précédemment, il ne devra attendre plus qu'une minute :



Ainsi, après trois minutes, tous les 4 fichiers seront entièrement téléchargés.

C'est de l'informatique !

La *barre de progression* (en anglais, *progress bar*) est un élément informatif qui permet d'indiquer à l'utilisateur l'état d'avancement d'une tâche que l'ordinateur est en train d'effectuer, comme par exemple l'installation d'un programme ou le téléchargement sur Internet. En règle générale, la barre de progression nous informe également sur la durée probable de l'accomplissement de la tâche en cours. Dans des cas particuliers, quand il n'est pas possible d'indiquer le temps restant, par exemple lors du téléchargement d'un fichier dont on ne connaît pas encore la taille totale, mais qu'on veut toutefois indiquer que l'ordinateur est en train d'accomplir la tâche, le curseur de la souris affichera soit un sablier, soit une icône tournante qui pourra avoir différents aspects.

Le débit de transfert des données, appelé aussi vitesse de transmission ou « bande passante » (un terme qui n'est pas tout à fait correct), correspond à la quantité des données numériques transmise via un canal de communication dans un intervalle de temps donné. La vitesse maximale du débit de transfert est appelée capacité de canal. Pour chaque canal de transmission, il existe en outre d'autres mesures importantes comme par exemple le délai de réponse appelé aussi la latence. Il désigne le temps nécessaire à un paquet de données pour passer d'un transmetteur à un récepteur (parfois, cette mesure comporte également le temps de parcours des données dans le sens inverse).

Sur Internet, il y a plusieurs facteurs qui influencent la vitesse maximale du débit de transfert : d'abord, un serveur ne peut fournir des données qu'à une vitesse maximale qui dépend de sa configuration matérielle et logicielle. Ensuite, ce serveur est connecté à Internet par une connexion disposant d'un débit de transfert de données limité et l'ensemble des processus du serveur qui transmettent des données doivent se partager cette connexion. De plus, les données sont transmises sur Internet via différentes lignes qui parcourent parfois le monde entier. Ces lignes sont très performantes, mais elles doivent transmettre en même temps un très grand nombre d'autres données qui circulent sur Internet. Un autre goulot d'étranglement se situe dans le raccordement de l'utilisateur au réseau Internet mis à disposition par son fournisseur d'accès Internet : en général, sa vitesse de transmission ne correspond même pas au dixième de celle dont dispose le serveur. Finalement, les données doivent encore être transmises du routeur de l'utilisateur à son ordinateur. Si une connexion câblée est en général plus rapide que la connexion à Internet elle-même, une connexion sans-fil (W-LAN) peut



réduire la vitesse de transmission. La capacité d'un réseau sans fil est généralement plus restreinte que celle d'un réseau câblé (ethernet) et n'assure souvent qu'un dixième de sa vitesse.

Sites web et mots clés

téléchargement, barre de progression

- https://fr.wikipedia.org/wiki/Barre_de_progression
- https://de.wikipedia.org/wiki/Datenübertragungsrate#Beispiele_für_Datenübertragungsraten





36. Accumuler des points

Le jeu de réflexion suivant est actuellement très populaire. On a besoin d'un tableau comme dans la figure ci-dessous. Le jeu démarre dans la case D (pour départ) et se termine dans la case A (pour arrivée). Le joueur avance en passant par les cases, et ceci, comme le montrent les deux petites flèches, toujours en respectant deux directions possibles : soit vers la droite soit vers le haut. Le but du jeu est de ramasser le plus de points possible : quand on passe par une case, on ramasse le nombre de points qui correspond. Pour accumuler un maximum de points, il faut donc choisir un parcours qui mène de D à A et qui passe par les cases avec le plus grand nombre de points.

	2	0	1	1	A
	1	2	0	2	3
	2	2	0	2	1
	3	1	0	2	0
↑	D	0	1	3	0
				→	

Prenons le tableau ci-dessus : quel est le nombre maximal de points qu'un joueur peut accumuler en parcourant les cases de D à A ?

- A) 10
- B) 12
- C) 14
- D) 16



Solution

La réponse C) est correcte.

Une des méthodes possibles pour trouver la solution est de noter progressivement le nombre maximal de points que l'on additionne en passant par les cases choisies. Commençons par la case tout en bas à gauche, soit par 0 :

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

Les chiffres en gras représentent le maximum de points que l'on peut accumuler. En avançant en haut, nous arrivons à 3 points (le chiffre 3 est en gras), en avançant à droite, le nombre de points reste zéro (le chiffre 0 est en gras) :

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

La case qui se trouve à droite de la case avec le chiffre 3 en gras et en haut de la case avec le chiffre 0 en gras ne peut être atteinte que de bas en haut ou de gauche à droite. Comme nous envisageons d'accumuler un maximum de points, nous avançons de gauche à droite et nous obtenons 4 points (0+3+1).

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	4	0	2	0
0	0	1	3	0

C'est en procédant ainsi que l'on peut accumuler le nombre maximal de points pour chaque case en additionnant le maximum de la case de gauche ainsi que de la case d'en bas avec la valeur de la case même. En termes mathématiques, on peut exprimer ce procédé de manière suivante :

$$v(i, 0) = 0$$

$$v(0, j) = 0$$

$$v(i, j) = c(i, j) + \max\{v(i - 1, j), v(i, j - 1)\}$$

à condition que $v(i, j)$ corresponde au maximum de la case (i, j) et que $c(i, j)$ corresponde à la valeur initiale de la case (i, j) .

Comme cette formule mathématique intègre toujours la case de gauche et d'en bas, il faut ajouter une colonne de droite et une ligne de colonne en bas du tableau avec des zéros.

Si on applique cette formule de manière cohérente, on arrive au résultat suivant :



0	8	9	10	12	14
0	6	9	9	11	14
0	5	7	7	9	10
0	3	4	4	6	6
0	0	0	1	4	4
0	0	0	0	0	0

C'est de l'informatique !

Il est très difficile et souvent assez pénible de trouver la «meilleure» solution parmi toutes les solutions possibles. Dans notre problème, on pourrait essayer tous les parcours possibles pour arriver à une solution possible. Une telle méthode s'appelle *force brute*. Pourtant, dans notre cas précis, appliquer une telle méthode signifierait d'analyser 70 parcours divers !

Comme nous venons de l'expliquer un peu plus haut, une méthode plus efficace serait de noter des solutions partielles. Cette technique s'appelle *mémoïsation* (le fait que la lettre «r» manque n'est pas une erreur de frappe) et en informatique, cette technique mémorise les valeurs retournées de la fonction *programmation dynamique*. Ainsi, il ne faut plus calculer que 25 valeurs.

Dans notre cas, cependant, nous pouvons également trouver des solutions partielles très avantageuses qui nous permettront d'accéder assez vite à la meilleure solution. Le tableau est très petit, c'est la raison pour laquelle on arrive très vite à rejeter toutes les autres solutions moins bonnes.

Sites web et mots clés

programmation dynamique, mémoïsation

- https://fr.wikipedia.org/wiki/Programmation_dynamique
- <https://fr.wikipedia.org/wiki/Mémoïsation>





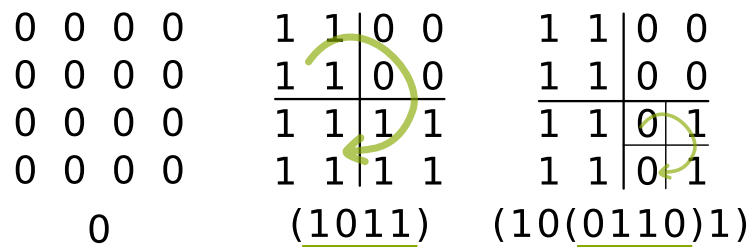
37. La méthode Quadtree

Il est possible de représenter des images en noir et blanc sous forme de caractères binaires 0 et 1 : chaque point de l'image est un pixel, 0 désigne un pixel blanc, 1 désigne un pixel noir. Une image au nombre de pixels 4×4 est donc représentée par 16 caractères binaires.

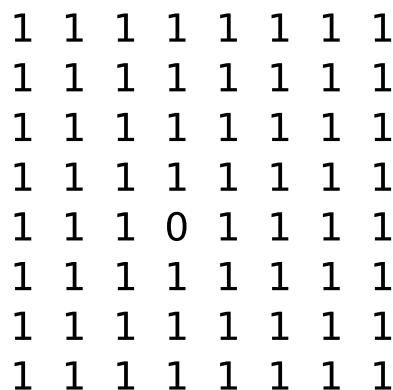
Beaucoup d'images peuvent toutefois être représentées par moins de caractères si on utilise la méthode *des quarts*. Pour ce faire, les caractères binaires sont disposés sur une grille carrée. Puis, on applique la méthode *des quarts* comme suit :

Si une grille n'est composée que d'un seul caractère binaire, le résultat sera exactement ce caractère. Autrement dit, si tous les caractères binaires d'une grille correspondent à 0, le résultat sera 0 (voir l'image à gauche). Si tous les caractères binaires de la grille correspondent à 1, le résultat sera 1. Si ce n'est pas le cas, on subdivisera ultérieurement la grille en quatre grilles plus petites de taille identique.

Cette méthode est alors appliquée à toutes les grilles partielles, toujours de gauche en haut à droite en bas et dans le sens des aiguilles d'une montre. Les quatre résultats (partiels) de chaque grille seront ensuite notés un après l'autre, entre parenthèses « (...) » (voir l'image au centre et à droite). Le résultat final est la séquence de caractères que l'on a ainsi obtenue.



Applique la méthode des quarts à l'image au nombre de pixels 8×8 suivante. Quel est le résultat final ?



- A) (1110)
- B) (11(1011)1)
- C) (111(1(1101)11))
- D) (111(1(1011)11))



Solution

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
<u>(111(1(1011)11))</u>							

On applique la méthode *des quarts* comme suit (voir aussi l'image à droite) : au début, la grille n'est pas composée de caractères identiques. C'est la raison pour laquelle nous subdivisons la grille en quatre grilles partielles. Le résultat aura donc la forme suivante :

$$(\langle \text{résultat 1} \rangle \langle \text{résultat 2} \rangle \langle \text{résultat 3} \rangle \langle \text{résultat 4} \rangle)$$

Les caractères des grilles partielles n° 1, n° 2 et n° 3 sont identiques : ils correspondent à 1. Le résultat de ces grilles est 1. Le résultat final aura donc la forme suivante :

$$(111 \langle \text{résultat 4} \rangle)$$

Comme la grille partielle n° 4 n'est pas composée de caractères identiques, nous devons la subdiviser. Dans les grilles partielles n° 4.1, n° 4.3 et n° 4.4, tous les caractères sont identiques et correspondent à 1. Ainsi, nous savons que le résultat final aura la forme suivante :

$$(111(1 \langle \text{résultat 4.2} \rangle 11))$$

Finalement, nous devons subdiviser la grille partielle n° 4.2. Dès lors, nous obtenons des caractères individuels. Le résultat de cette dernière grille partielle est (1011) et le résultat final sera donc :

$$(111(1(1011)11))$$

C'est de l'informatique !

Nous utilisons de plus en plus de systèmes informatiques comme des ordinateurs, des smartphones, des ordinateurs portables ou des tablettes. Par conséquent, nous produisons de plus en plus de données numériques. Même si les systèmes offrent toujours davantage de capacités de stockage et de transmission des données numériques, cela ne suffit pas. Il est donc intéressant de réfléchir aux méthodes qui permettent de compresser des textes, des images ou des données audio et vidéo de façon optimale. Compresser des données informatiques veut dire que l'on économise de la mémoire en réduisant la taille des informations par des algorithmes mathématiques. Certaines méthodes de compression sont dites «sans perte», c'est-à-dire que l'on ne perd aucune information, alors que d'autres sont «avec perte», c'est-à-dire que l'on va perdre un peu de l'information originale au cours du processus de compression. Naturellement, sur une image, une telle perte est peu visible, sur un fichier de données, par contre, ce sera destructeur.

L'informatique connaît beaucoup de méthodes de compression. La méthode décrite dans la présente tâche fonctionne particulièrement bien s'il y a de grandes parties de l'image qui sont composées de données identiques. La subdivision en quatre parties identiques (et si nécessaire en quatre sous-parties additionnelles et ainsi de suite) peut être comprise à l'aide de cette méthode des quarts appelée aussi *Quadtree* ou *arbre quaternaire*. Un arbre quaternaire est une structure de données de type arbre. Cette structure de données joue un rôle très important dans le domaine de l'informatique.



Sites web et mots clés

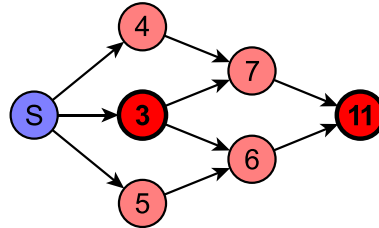
quadtree, compression d'image, arbre quaternaire
— <https://fr.wikipedia.org/wiki/Quadtree>





38. Raccourci ou détour?

L'image ci-dessous représente le plan d'une ville montrant les rues à sens unique. Les chiffres et les nombres affichés sur les carrefours nous informent sur la longueur du parcours minimal entre le point «S» et le carrefour correspondant.



Observons les deux carrefours marqués en rouge : laquelle des affirmations suivantes est valable par rapport à ces deux carrefours ?

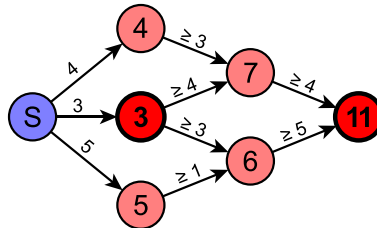
- A) La longueur du parcours minimal entre ces deux carrefours est exactement de 8.
- B) La longueur du parcours minimal entre ces deux carrefours est de 8 ou moins de 8.
- C) La longueur du parcours minimal entre ces deux carrefours est de 8 ou plus de 8.
- D) On ne peut rien dire de précis sur la longueur du parcours minimal entre ces deux carrefours.



Solution

La réponse correcte est C) : «La longueur du parcours minimal entre ces deux carrefours est de 8 ou plus de 8.»

Si la longueur du parcours minimal était inférieure à 8, la longueur du parcours minimal entre «S» et le carrefour marqué par le nombre «11» serait inférieure à $3 + 8 = 11$. Le parcours minimal pourrait très bien être supérieur à 8, puisque la longueur du parcours minimal entre «S» et le carrefour marqué par le nombre «11» pourrait passer par les carrefours marqués par les chiffres «4» et «7» ou «5» et «6». C'est pourquoi les autres réponses ne sont pas correctes.



L'image ci-dessus montre la longueur minimale pour chaque parcours afin de justifier les chiffres et nombres affichés sur chaque carrefour.

C'est de l'informatique !

Dans un premier temps, on pourrait croire que le parcours minimal passe à travers les deux carrefours en question. Si tel était le cas, nous devrions connaître la longueur effective de la distance entre ces deux carrefours. Malheureusement, nous ne la connaissons pas. En effet, dans notre tâche, nous ne connaissons que la longueur du parcours minimal entre le point «S» et le carrefour en question et non pas la longueur effective de la distance entre ces deux points. Il se peut que la longueur de la distance entre «4» et «7» soit 10 si le parcours de «3» à «7» est 4. C'est la raison pour laquelle nous pouvons seulement affirmer que le parcours minimal entre les deux carrefours en question est plus grand que 8 ou égal à 8.

Sites web et mots clés

parcours minimal

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



39. L'affichage numérique

Un affichage à 7 segments représente les chiffres de la manière suivante :



Chaque chiffre comporte 7 segments au maximum. Imaginons que nous avons un affichage à un seul chiffre et que l'affichage est partiellement bloqué.

Indique les segments qui ne doivent pas être cachés pour que l'affichage reste lisible.





Solution

La réponse correcte est :



Les paires de chiffres 1/7, 3/9, 5/6, 6/8 et 8/0 constituent la clé du problème. Ces paires de chiffres ne se distinguent que par un seul segment. En supprimant ce segment, on ne pourrait plus distinguer les chiffres d'une paire :



Les deux segments restants en bas et en bas à droite ne sont pas pertinents pour distinguer les paires de chiffres, car toutes les paires se distinguent par un autre segment au moins en plus de ces deux segments restants ou elles ont exactement ces deux segments en commun.

C'est de l'informatique !

On parle de redondance quand une partie d'une information peut être supprimée sans que le message ne devienne incomplet ou illisible. Les textes sont un bon exemple de redondance, car ils restent en général lisibles même si quelques petits changements y ont été effectués. En informatique, les fichiers redondants sont, certes, gourmands en espace-mémoire, mais d'un autre côté, la redondance les rend plus stables face à des modifications involontaires qui pourraient se produire, par exemple, lors de l'enregistrement ou de la transmission des données. En informatique, la redondance ne pose donc pas forcément de problème.

Pourtant, il se peut que l'on veuille éliminer la redondance dans un fichier. Cette procédure s'appelle «compression des données» et elle permet d'économiser de la mémoire ou de réduire la taille des informations pour une transmission plus rapide. Normalement, un fichier compressé reste lisible et compréhensible, mais quand un fichier compressé est endommagé ou corrompu, on n'arrive souvent plus à le récupérer. Aujourd'hui, beaucoup de types de fichiers sont compressés. Pour éviter des pertes de données, on ajoute parfois des données redondantes qui permettent de corriger des erreurs minimales.

Sites web et mots clés

identification des informations, compression, chiffres, affichage à 7 segments, redondance

— https://fr.wikipedia.org/wiki/Afficheur_7_segments



40. Subdivision du code

Dans un procédé de codage spécial qui permet de crypter des textes, chaque lettre est remplacée par un nombre composé des chiffres 0 à 9. La particularité de ce cryptage réside dans le fait que chaque chiffre ou nombre qui remplace une lettre ne doit pas commencer par le chiffre ou nombre associé à une autre lettre.

Voilà quelques exemples : la lettre «X», par exemple, est remplacée par le nombre 12. La lettre «Y» peut donc, par exemple, être remplacée par le chiffre 2 car le nombre 12 ne commence pas par le chiffre 2 (et le chiffre 2 ne commence pas par 12). La lettre «Z» peut ainsi être remplacée par le nombre 11, car ni le nombre 12 ni le chiffre 2 ne commencent par 11, et le nombre 11 ne commence ni par 12 ni par 2. Cependant, il ne serait pas possible de remplacer la lettre «Z» par le nombre 21 car ce dernier commence par le chiffre 2 qui, lui, remplace déjà la lettre «Y».

Le mot BEBRAS vient d'être crypté par la suite de chiffres 12112233321. Laquelle des subdivisions suivantes représente les lettres du mot BEBRAS ?

- A) 12 11 22 33 32 1
- B) 1 21 1 22 33 321
- C) 1 21 12 2 33 321
- D) 1 21 1 22 3 3321
- E) 12 1 12 23 33 21



Solution

La réponse correcte est :

1 21 1 22 33 321

Pour expliquer la solution, nous commençons à l'extrême gauche, soit au début de la suite de chiffres : si l'on remplaçait la lettre «B» par le nombre 12, la lettre «E» devrait être obligatoirement remplacée par le chiffre 1 (car le chiffre 1 est suivi par le nombre 12 qui, lui, remplace la lettre B). Si l'on procédait ainsi, on contredirait la règle qui détermine les préfixes parce que la lettre «B» commencerait par le chiffre 1 qui, lui, remplace la lettre «E». Ensuite, il n'est pas possible de remplacer la lettre «B» par un nombre qui comporte plus de 2 chiffres (comme par exemple 121, 1211, 12112, etc.) car ce nombre devrait être répété à l'intérieur d'une des suites de chiffres, ce qui n'est le cas d'aucune d'entre elles. Par conséquent, la lettre «B» est forcément remplacée par le chiffre 1.

La lettre «E» est suivie de la lettre «B». Il s'ensuit que la lettre «E» ne peut être remplacée que par le chiffre ou par les suites de chiffres suivants : 2, 21 ou 211223332. Selon la règle préalablement établie, il n'est pas possible de remplacer la lettre «E» par le chiffre 2 car le mot commencerait ainsi par la suite de lettres suivante : BEBB. Il n'est également pas possible de remplacer la lettre «E» par 211223332 car le mot complet serait alors BEB. Il en résulte que la lettre «E» a été remplacée par le nombre 21 et c'est ainsi que nous savons que la suite de lettres BEB est remplacée par la suite de chiffres suivante : 1 21 1. Ce qui nous reste maintenant à faire, c'est de subdiviser la suite de chiffres 2233321.

Examinons ensuite la lettre «S» à la fin du mot. Elle ne peut être remplacée ni par 1 ni par 21 parce que le chiffre 1 et le nombre 21 remplacent déjà les lettres «B» et «E». Voilà les seules suites de chiffres possibles qui puissent remplacer la lettre «S» : 321, 3321, 33321, 233321 et 2233321. Mais, si l'on remplaçait la lettre «S» par la suite de chiffres 2233321, le résultat complet serait BEBS. De même, il n'est pas possible de remplacer la lettre «S» par la suite de chiffres 233321 parce qu'il ne resterait qu'un seul chiffre pour remplacer les lettres «R» et «A». Il n'est également pas possible de remplacer la lettre «S» par la suite de chiffres 33321 car les deux chiffres identiques qui composent le nombre 22 restant devraient remplacer deux lettres différentes, la lettre «R» et la lettre «A». Si «S» était remplacé par la suite de chiffres 3321, les deux lettres «R» et «A» seraient représentées par 223. Le problème qui se pose ici est que le chiffre 2 ne pourrait pas remplacer la lettre «R» et le chiffre 3 ne pourrait pas remplacer la lettre «A», parce que ces deux chiffres sont déjà associés à d'autres lettres. Pour remplacer la lettre «S» il ne reste donc que la suite de chiffres 321. Finalement, il reste à subdiviser la dernière suite de chiffres qui est : 2233. Pour les raisons mentionnées ci-dessus, le nombre 22 remplacera la lettre «R» et le nombre 33 remplacera la lettre «A».

C'est de l'informatique !

Le processus de cryptage décrit dans notre exercice illustre le codage à travers un code *préfixe*. En général, le préfixe est une suite d'éléments qui précède une autre suite d'éléments (comme par exemple des caractères, des chiffres, des lettres, et ainsi de suite). Par analogie, en informatique, le code préfixe est un code dont la particularité est la suivante : le code ne possède aucun mot ayant pour préfixe un autre mot.

Les codes préfixes comportent des mots de longueur variable. Grâce à la règle présentée dans notre exercice, les codes préfixes permettent de voir tout de suite quand un mot commence et quand il se termine, sans qu'il soit nécessaire d'utiliser des séparateurs. De plus, en associant des codes courts à des lettres fréquemment utilisées, on arrive à crypter des textes de manière très efficace et par conséquent, à économiser de la mémoire.



Le codage Huffman permet de trouver un code préfixe optimal. Son utilisation est très fréquente, en particulier pour la compression des fichiers JPEG ou MP3.

Sites web et mots clés

code préfixe, cryptographie, cryptanalyse, déchiffrer un message codé

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



A. Auteurs des exercices

 Andrea Adamoli	 Arnheiður Guðmundsdóttir	 J.P. Pretti
 Nursultan Akhmetov	 Yasemin Gülbahar	 Daniel Rakijašić
 Adil Aliyev	 Martin Guggisberg	 Chris Roffey
 Haim Averbuch	 Urs Hauser	 Frances Rosamond
 Khuyagbaatar Batsuren	 Hans-Werner Hein	 Kirsten Schlüter
 Wilfried Baumann	 Fredrik Heintz	 Victor Schmidt
 Bartosz Bieganski	 Juraj Hromkovič	 Eljakim Schrijvers
 Daphne Blokhuis	 Ungyeol Jung	 Masood Seddighin
 Eugenio Bravo	 Filiz Kalelioğlu	 Maiko Shimabuku
 Andrej Brodnik	 Dong Yoon Kim	 Taras Shpot
 Carmen Bruni	 Vaidotas Kinčius	 Martin Stangl
 Amaury A. Castro Jr.	 Jia-Ling Koh	 Seiichi Tani
 Marios Choudary	 Ivana Kosírová	 Gergely Tassy
 Anton Chukhnov	 Regula Lacher	 Monika Tomcsányiová
 Raluca Constantinescu	 Greg Lee	 Peter Tomcsányi
 Zsófia Csepregi-Horváth	 Milan Lukić	 Ahto Truu
 Valentina Dagienė	 Dario Malchiodi	 Willem van der Vegt
 Darija Dasović Rakijašić	 Hiroki Manabe	 Jiří Vaníček
 Christian Datzko	 Dimitris Mavrovouniotis	 Troy Vasiga
 Susanne Datzko	 Mattia Monga	 Nicolette Venn
 Janez Demšar	 Henry Ong	 Corina Elena Vint
 Olivier Ens	 Zsuzsa Pluhár	 Michael Weigend
 Hanspeter Erni	 Wolfgang Pohl	 Hongjin Yeh
 Michael Fellows	 Ilya Posov	 Momo Yokoyama
 Gerald Futschek	 Sergei Pozdniakov	 Khairul A. Mohamad Zaki



B. Sponsoring : Concours 2017

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

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

d digitec.ch

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



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



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



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/>



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.

n|w

Fachhochschule
Nordwestschweiz

<https://www.fhnw.ch/de/die-fhnw/hochschulen/ph>
Pädagogische Hochschule FHNW

Z

hdk

Zürcher Hochschule der Künste
Game Design

<https://www.zhdk.ch/>
Zürcher Hochschule der Künste



ZUBLER & PARTNER AG
Informatik

<http://www.zubler.ch/>
Zubler & Partner AG Informatik

senarclens
leu+partner
strategische kommunikation

<http://senarclens.com/>
Senarclens Leu & Partner



C. Offres ultérieures

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischerverein für informatik und
erausbildung // société suisse de l'inform
atiquedans l'enseignement // società sviz
zera per l'informatica nell'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.