INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Exercices et solutions 2017
Années scolaires 9/10

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

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

SSIE
www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissedel'inform
atiquedansl'enseignement//societàsviz
zeraperl'informaticanell'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 LaTeX.

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.bebras.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 fausse réduisait le total des points. Ne pas répondre à une question n'avait aucune incidence sur le nombre de points. Le nombre de points de chaque exercice était fixé en fonction du son degré de difficulté :

	Facile	Moyen	Difficile
Réponse correcte	6 points	9 points	12 points
Réponse fausse	−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.

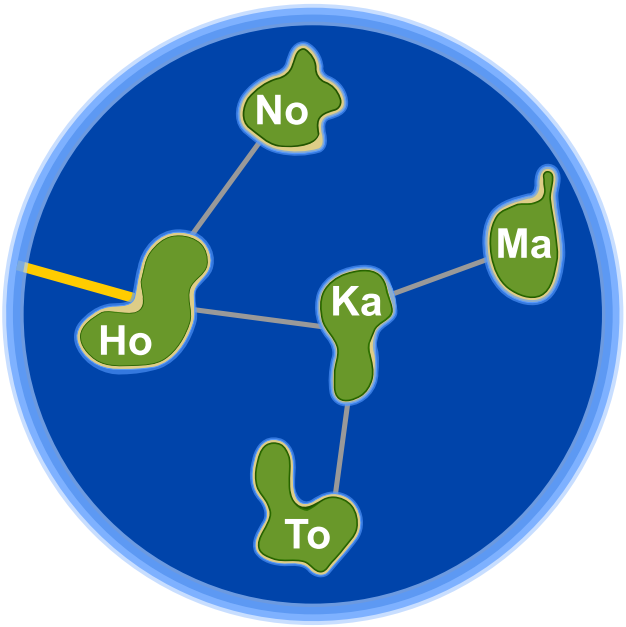
Pour de plus amples informations :

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

1. 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.



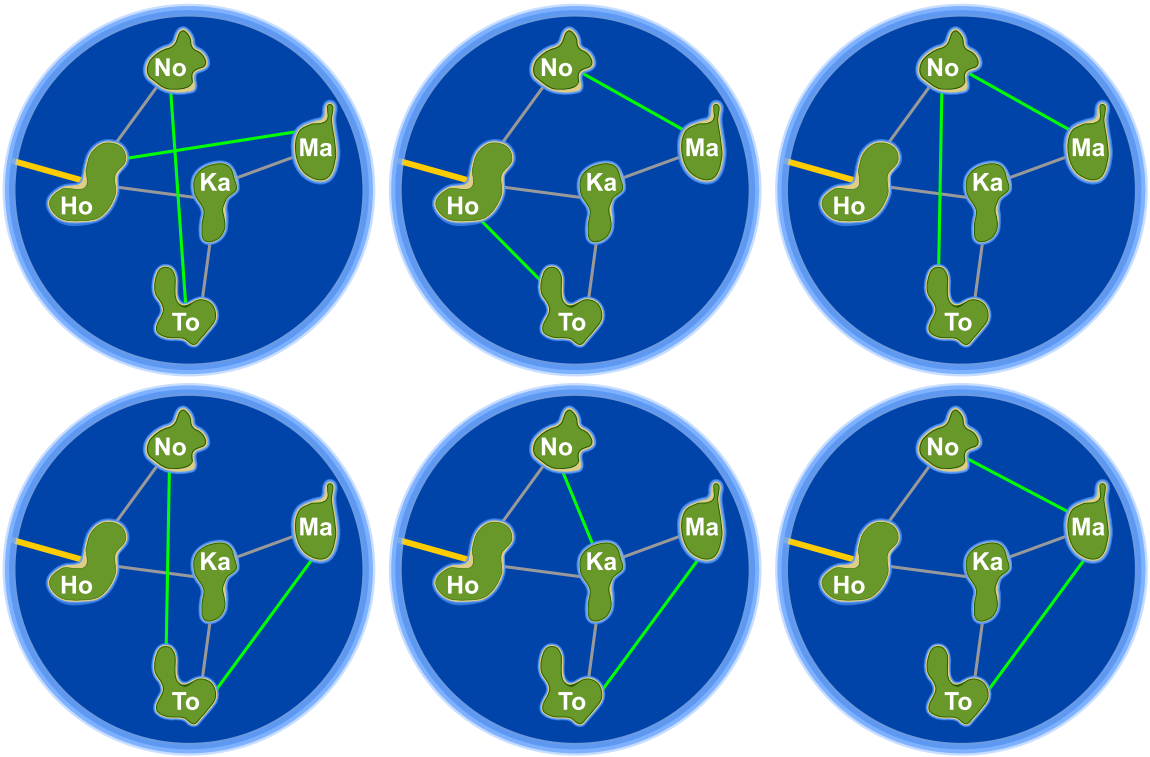
Les habitants de Honomakato demandent une connexion fiable à Internet pour toutes les îles : cela veut dire que même si un des câbles est endommagé, chacune des îles doit avoir accès à Internet.
Fais en sorte que l'archipel Honomakato obtienne une connexion fiable à Internet. Pose deux autres câbles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

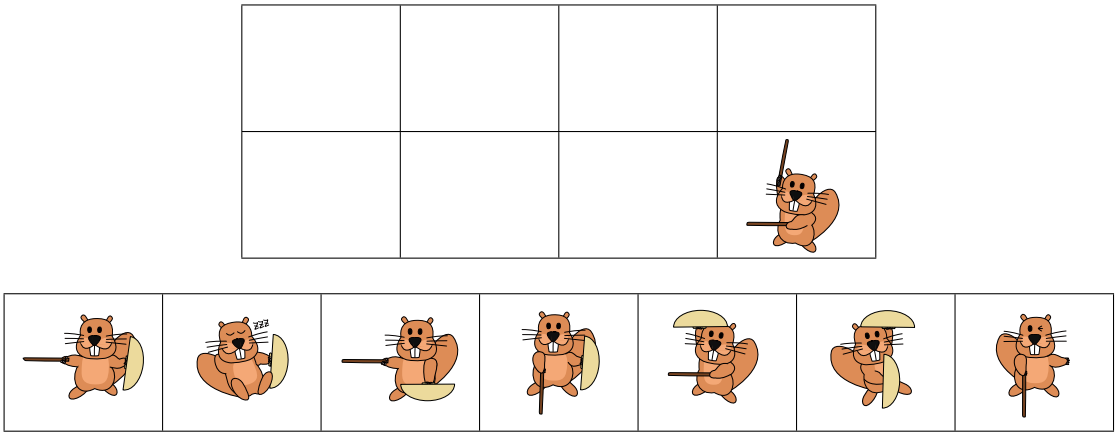
Sites web et mots clés

structure de données dynamique, graphe, pont (informatique)

— `https://fr.wikipedia.org/wiki/Séparateur_(théorie_des_graphes)`

2. Un art martial japonais

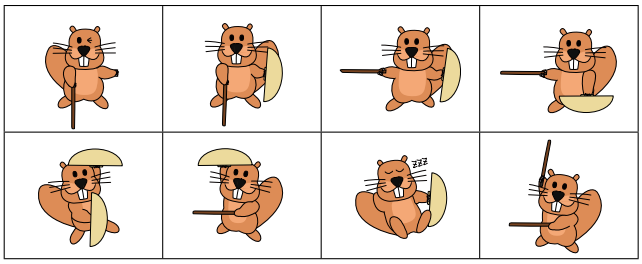
Lucia et ses amis sont membres d'un club d'art martial japonais qui enseigne le maniement du bâton. Pour une photo dans la cour de récréation, ils aimeraient bien se mettre en place afin que chaque bâton vise un bouclier. Pour que chacun puisse se mettre correctement en place, on a dessiné quelques cases 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

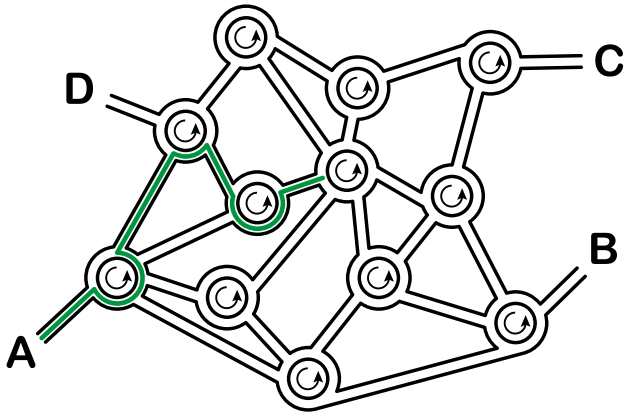
- `https://fr.wikipedia.org/wiki/Retour_sur_trace`
- `http://www.inf-schule.de/grenzen/komplexitaet/affenpuzzle/einstieg_affenpuzzle`

3. 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 rond-point, prends la quatrième sortie.
— Au prochain rond-point, prends la première sortie.
— Au prochain rond-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.

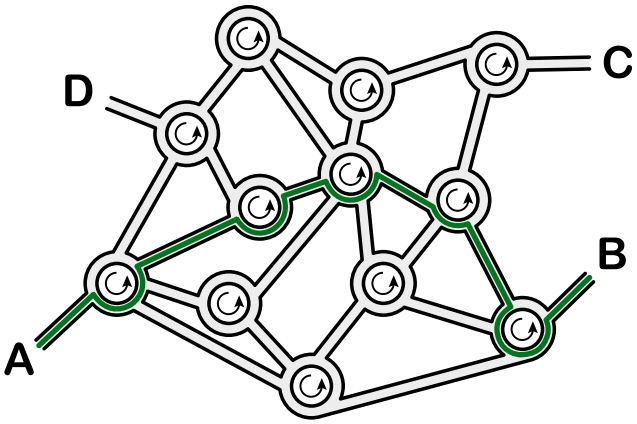
B) L'indication le mènera au point B.

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, ils 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.

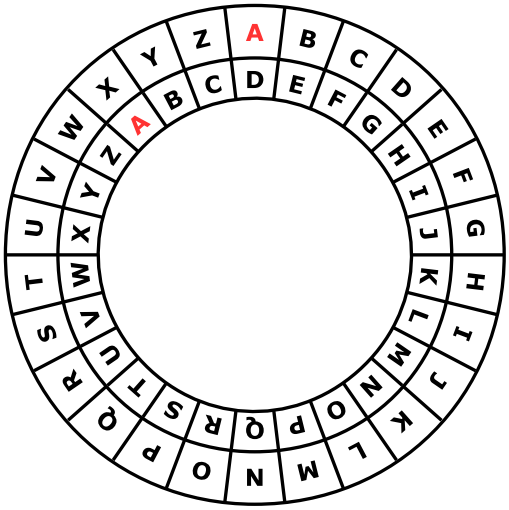
Sites web et mots clés

séquences, exécution de programmes, langage formel

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

4. 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.

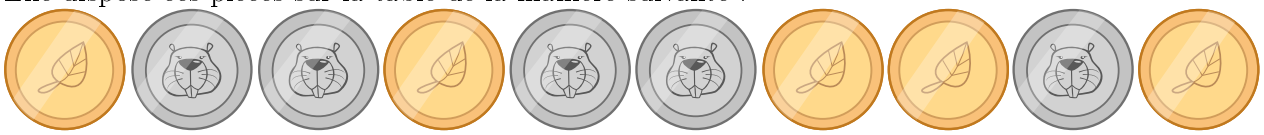
Sites web et mots clés

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`

5. 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.

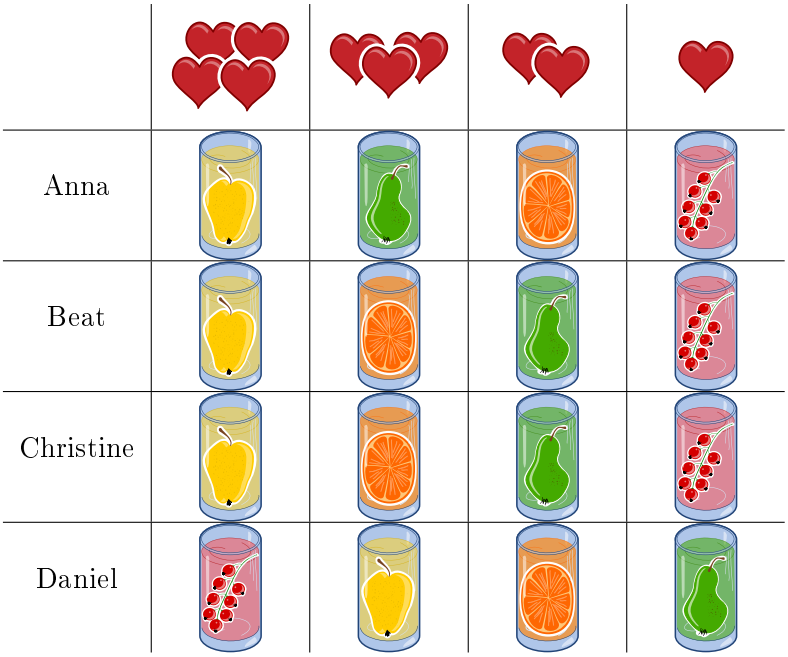
Sites web et mots clés

parité, bit de parité

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

6. Bar à jus de fruits

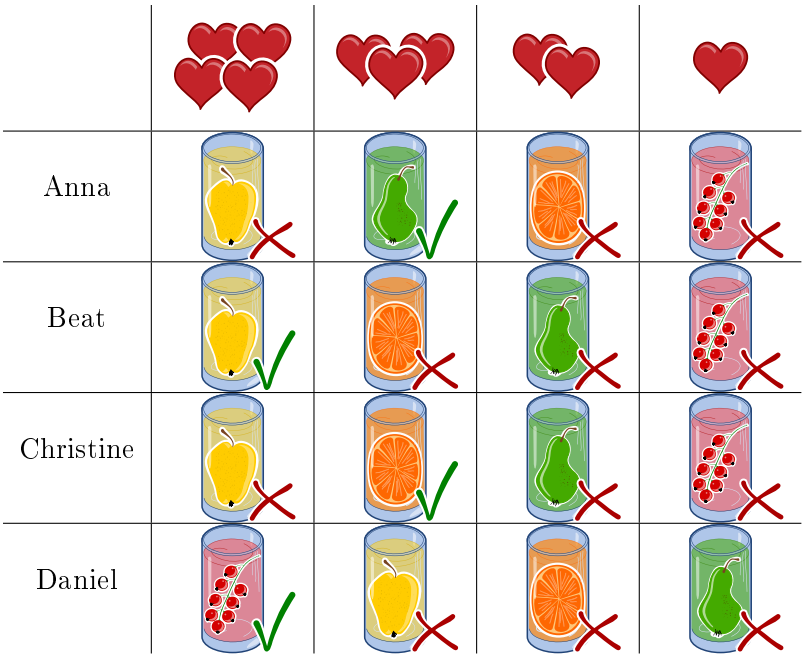
Sur leur route de vacances, quatre amis font une halte pour se rafraîchir dans un bar à jus de fruits. Chacun d'entre eux a ses propres préférences en ce qui concerne la saveur des jus. Celles-ci sont représentées dans le tableau ci-dessous. Plus il y a de cœurs, plus la personne en question préfère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



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 :



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.

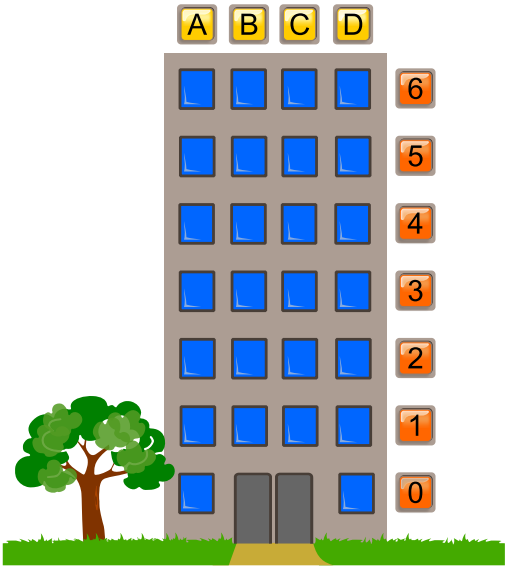
Sites web et mots clés

optimisation, couplage

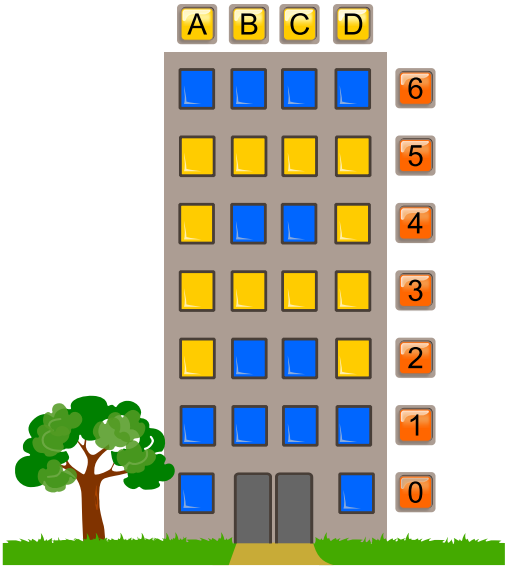
- `https://fr.wikipedia.org/wiki/Optimisation_(mathématiques)`
- `https://fr.wikipedia.org/wiki/Séparation_et_évaluation`
- `https://fr.wikipedia.org/wiki/Couplage_(théorie_des_graphes)`

7. 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 billiards 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`

8. 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)

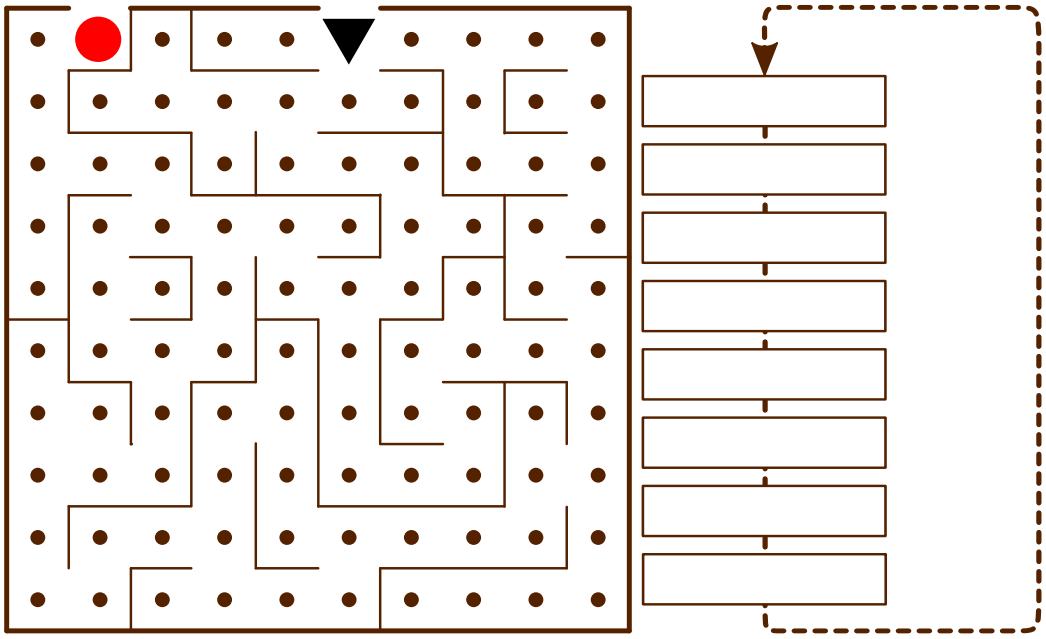
— `https://fr.wikipedia.org/wiki/Grammaire_formelle`

9. 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 :

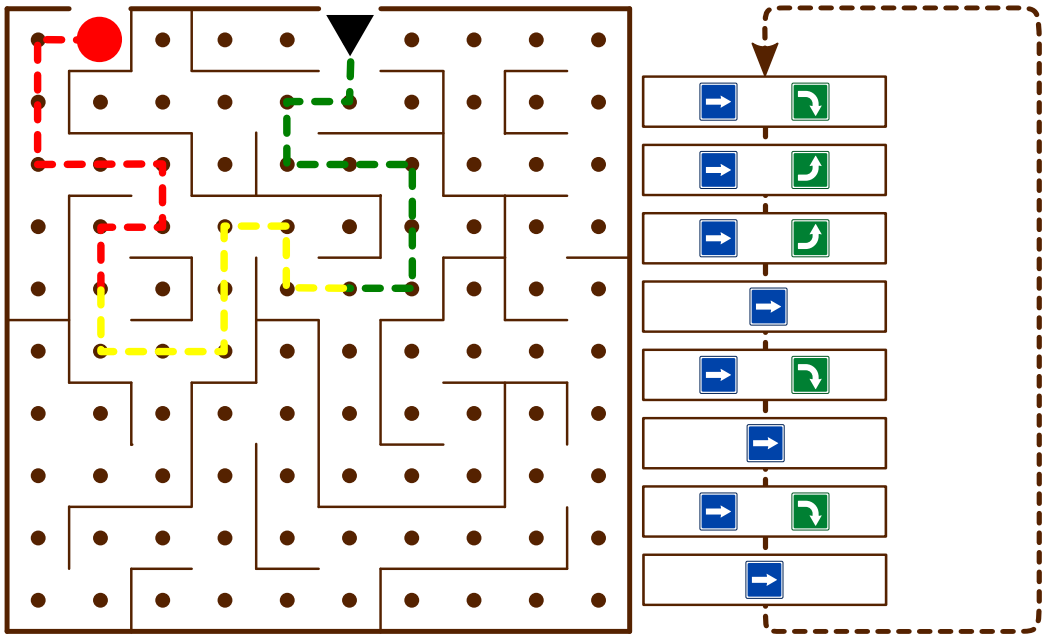
	Fais un pas en avant, ensuite, tourne à gauche.
	Fais un pas en avant, ensuite, tourne à droite.
	Fais un pas en avant.

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.

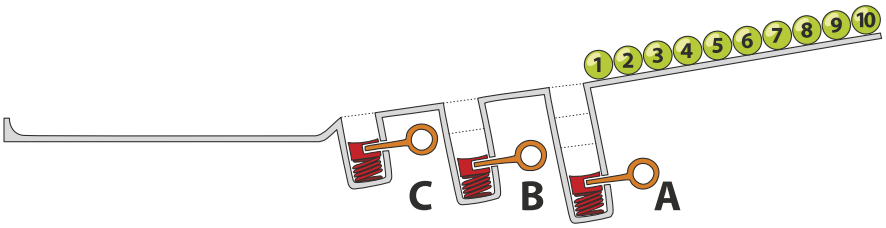
Sites web et mots clés

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

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

10. 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)

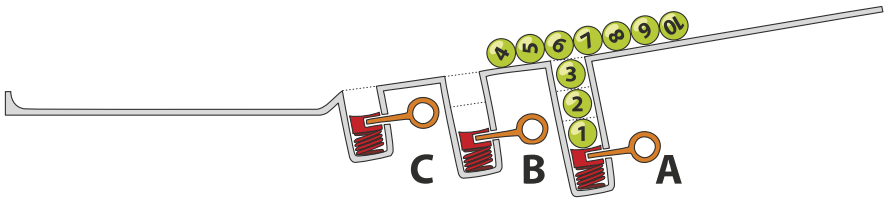

B)


C)

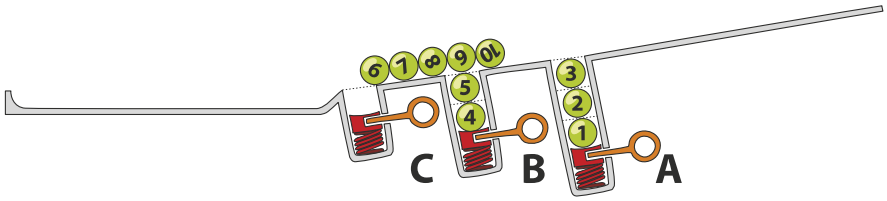

D)


Solution

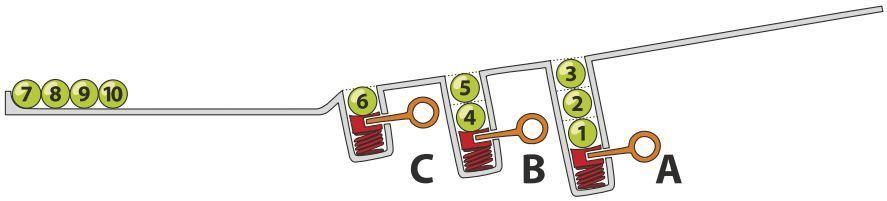
La réponse correcte est D)  .
Les billes 1, 2 et 3 tombent dans le trou A, les billes 4 à 10 passent le trou A (qui, lui, contient les billes 1 à 3).



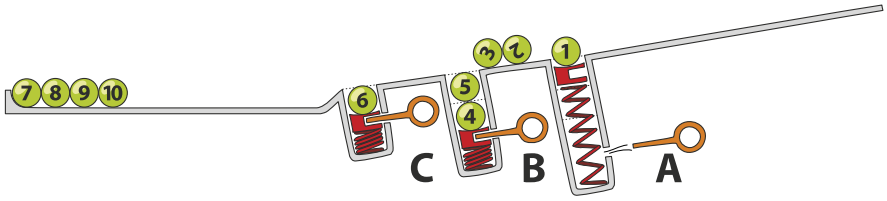
Ensuite, les billes 4 et 5 tombent dans le trou B et les billes 6 à 10 passent le trou B (qui, lui, contient les billes 4 à 5).



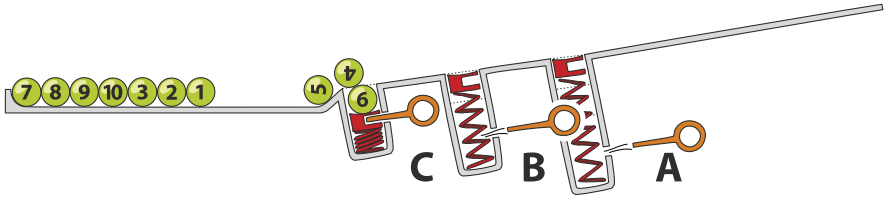
Finalement, la bille 6 tombe dans le trou C et les billes 7 à 10 passent le trou C (qui, lui, contient la bille 6). Ces-dernières atteignent la fin de la rampe dans l'ordre de 7 à 10.



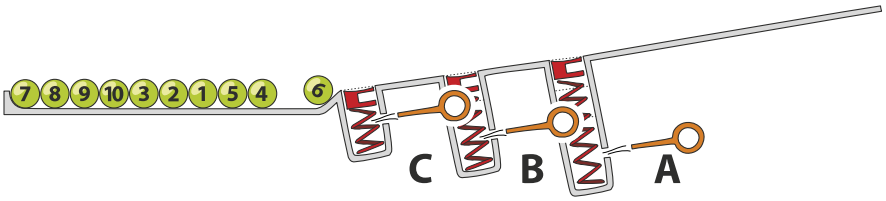
Quand les billes 7 à 10 ont passé les trous, le ressort du trou A libère les billes 3, 2, 1. Celles-ci se retrouvent donc de nouveau sur la rampe et roulent en direction de la fin de la rampe.



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 !

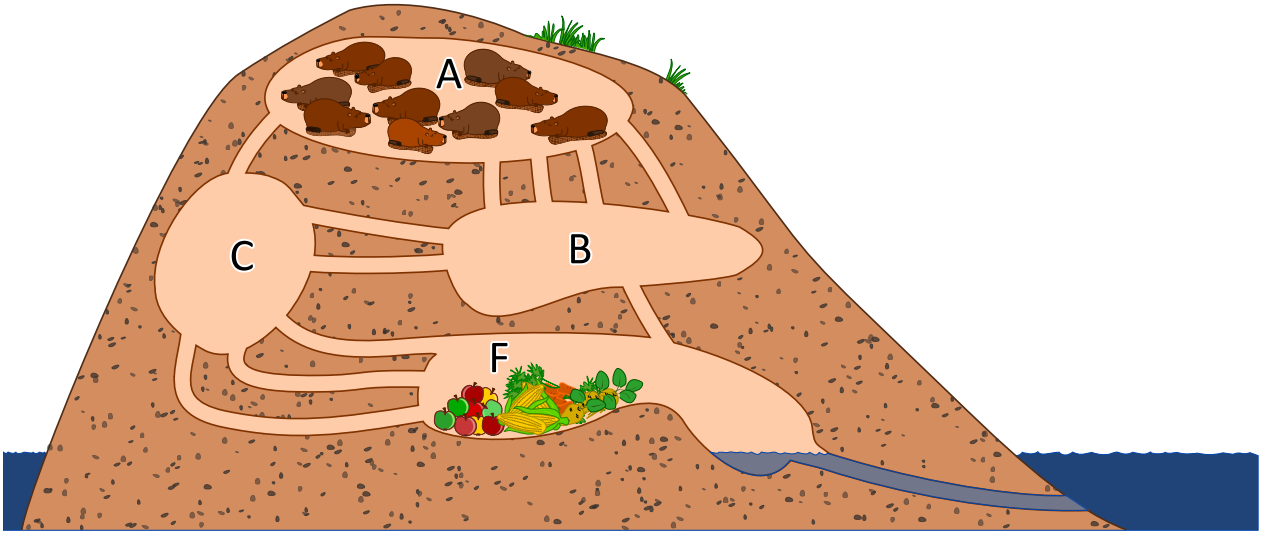
Les trous de la rampe rappellent la structure de la pile (en anglais, *stack*). En informatique, une pile permet d'enregistrer des données pour les réutiliser ensuite selon le principe de *Last-In First-Out (LIFO)* (en français «dernier arrivé, premier sorti») : ainsi, la dernière bille qui est tombée 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)`

11. 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
3 castors se déplacent de A à B (moins que ce qui serait possible) *1 castor se déplace de A à C*				
Situation après 1 minute	6	3	1	0
3 castors se déplacent de A à B (moins que ce qui serait possible) *1 castor se déplace de B à F* *2 castors se déplacent de B à C* *1 castor se déplace de C à F* *1 castor se déplace de A à C*				
Situation après 2 minutes	2	3	3	2
1 castor se déplace de A à B (le chemin le plus court) *1 castor se déplace de B à F* *2 castors se déplacent de B à C* *1 castor se déplace de A à C (le chemin le plus court)* *3 castors se déplacent de C à F*				
Situation après 3 minutes	0	1	3	6
1 castor se déplace de B à F *3 castors se déplacent de C à F*				
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.

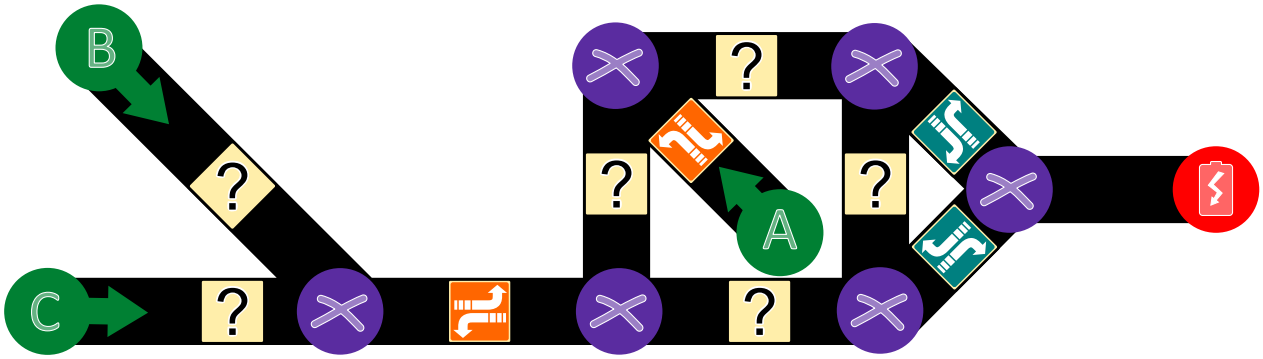
Sites web et mots clés

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`

12. Aide l'Arabot !

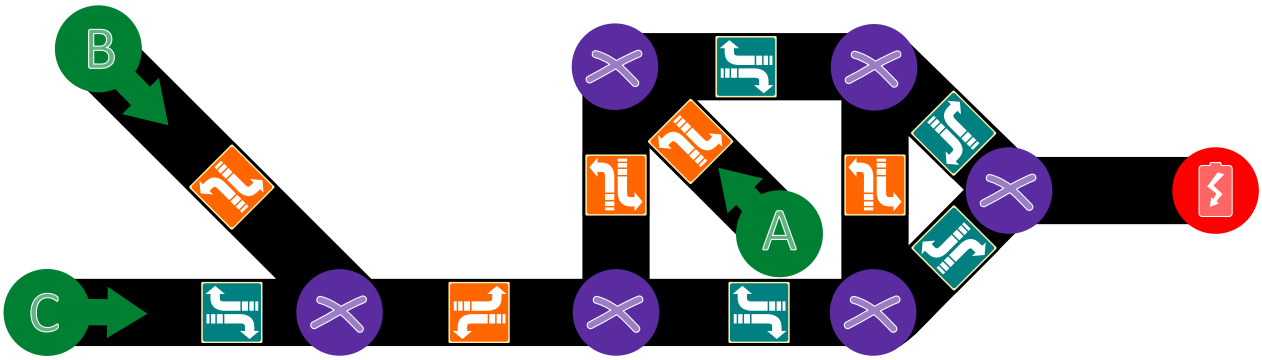
Un Arabot est un robot qui se déplace sur une feuille de papier. Il suit toujours les lignes noires qui se trouvent sur cette feuille. Toutes les lignes forment un parcours et chacune d'entre elles comporte une ou plusieurs instructions qui informent l'Arabot sur ce qu'il doit faire au prochain carrefour ( ) : soit prendre la première sortie à gauche ( ), soit prendre la première sortie à droite ( ). Quand il arrive au point A, B ou C, il ne sait plus comment continuer et il s'éteint. Il est souhaitable que l'Arabot démarre toujours au point A, B ou C et qu'il finisse son parcours à la station de recharge ( ).
Jonas a créé un nouveau parcours pour son robot. Son parcours comporte déjà quelques instructions, mais malheureusement, pour quelques lignes, il ne sait pas encore quelles instructions il doit insérer pour que son robot termine le parcours à la station de recharge.



Aide Jonas à insérer les instructions manquantes afin que son robot termine son parcours à la station de recharge.

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, médicine computationnelle

- `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`

13. 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`

14. 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 :

$$\text{Emil} \rightarrow \text{Eril} \rightarrow \text{Eric} \rightarrow \text{Erich}$$

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.

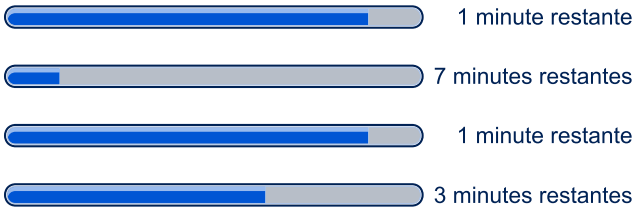
Sites web et mots clés

distance de Levenshtein, distance d'édition

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

15. 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'usager 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`

A. Auteurs des exercices

Andrea Adamoli
Wilfried Baumann
Bartosz Bieganowski
Daphne Blokhuis
Eugenio Bravo
Carmen Bruni
Anton Chukhnov
Zsófia Csepregi-Horváth
Valentina Dagienė
Christian Datzko
Susanne Datzko
Janez Demšar
Olivier Ens
Hanspeter Erni
Michael Fellows
Gerald Futschek
Martin Guggisberg
Urs Hauser
Juraj Hromkovič
Filiz Kalelioğlu
Vaidotas Kinčius
Ivana Kosírová
Regula Lacher
Greg Lee
Milan Lukić
Hiroki Manabe
Mattia Monga
Henry Ong
Wolfgang Pohl
Sergei Pozdniakov
Frances Rosamond
Kirsten Schlüter
Eljakim Schrijvers
Maiko Shimabuku
Taras Shpot
Seiichi Tani
Ahto Truu
Jiří Vaníček
Troy Vasiga
Michael Weigend
Hongjin Yeh
Momo Yokoyama

B. Sponsoring : Concours 2017

http://www.haslerstiftung.ch/

http://www.roborobo.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

Bits with Bite
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

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.