



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

Exercices et solutions 2017 Années scolaires 11/12/13

<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
erausbildung // 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. 42.



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>



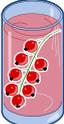
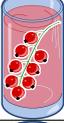
Table des matières

Ont collaboré au Castor Informatique 2017	i
Préambule	ii
1. Bar à jus de fruits	1
2. Substitutions	5
3. Sors du labyrinthe !	7
4. Système d'arrosage	9
5. Une nouvelle chanson	11
6. Jeu de billes	13
7. À table, mais vite !	17
8. Aide l'Arabot !	21
9. Les piles de cure-dents à diviser	23
10. Calculer la distance entre les mots	25
11. Accumuler des points	27
12. La méthode Quadtree	31
13. Raccourci ou détour ?	35
14. L'affichage numérique	37
15. Subdivision du code	39
A. Auteurs des exercices	42
B. Sponsoring : Concours 2017	43
C. Offres ultérieures	45



1. 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))





2. 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



3. 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.

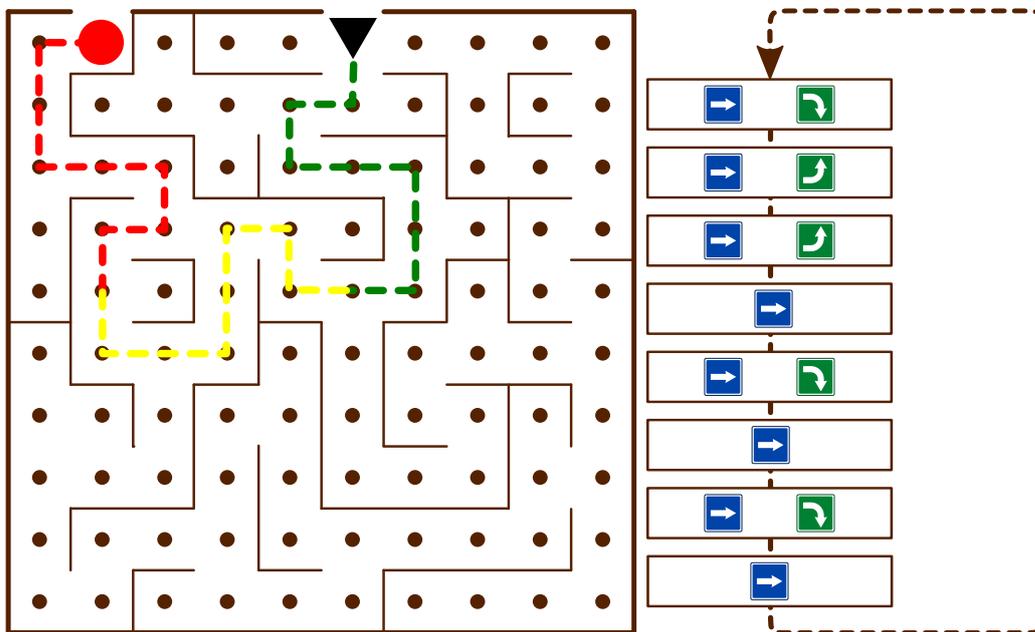
Même si Benjamin ne peut mémoriser qu'une série de huit instructions au maximum, il peut effectuer les différentes instructions de manière répétitive jusqu'à ce qu'il sorte du labyrinthe.

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

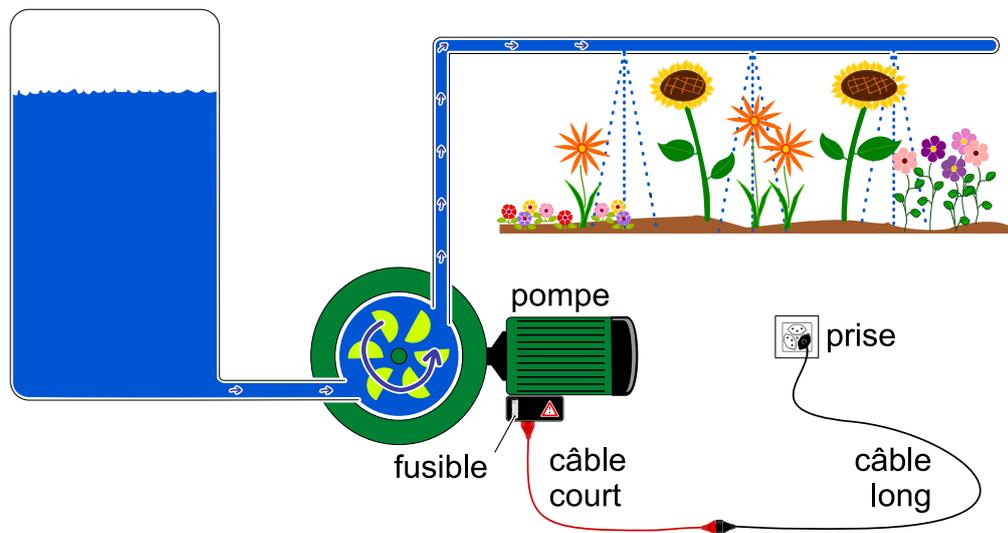


4. Système d'arrosage

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

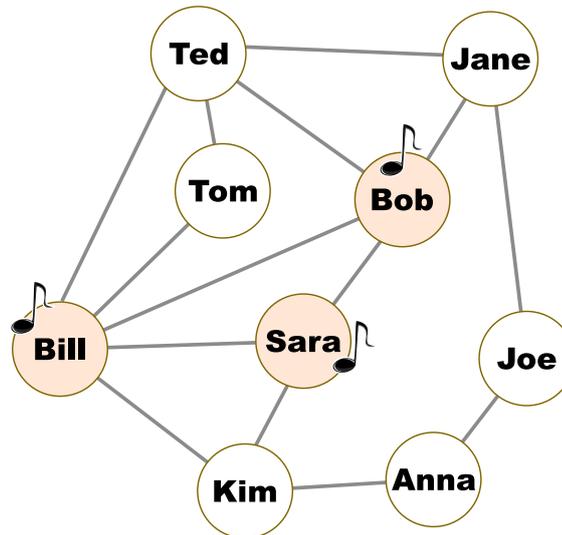
vérification, débogage

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



5. 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 ?

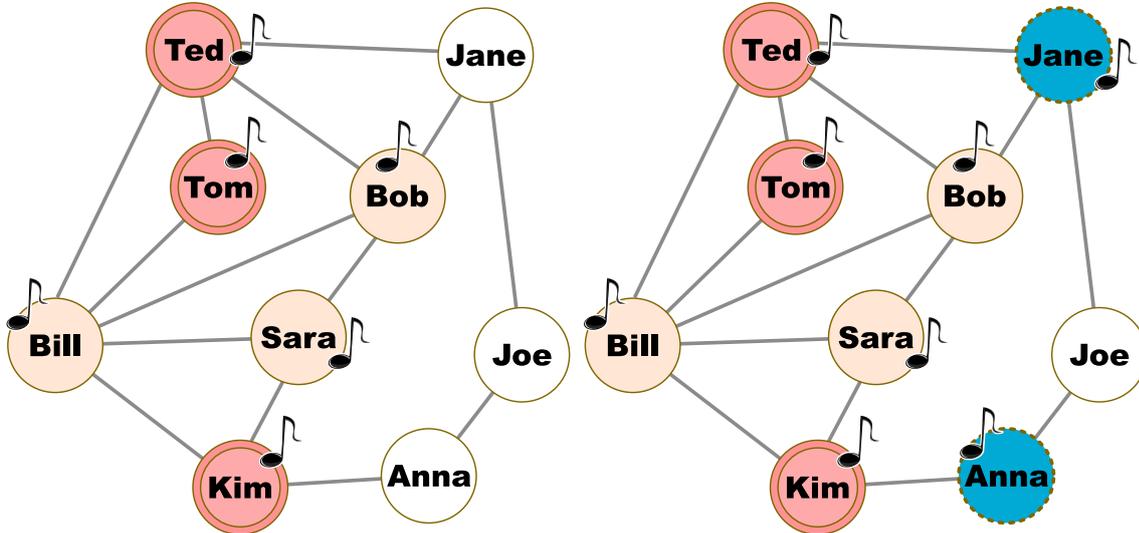
- A) Mercredi
- B) Jeudi
- C) Vendredi
- D) Samedi



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

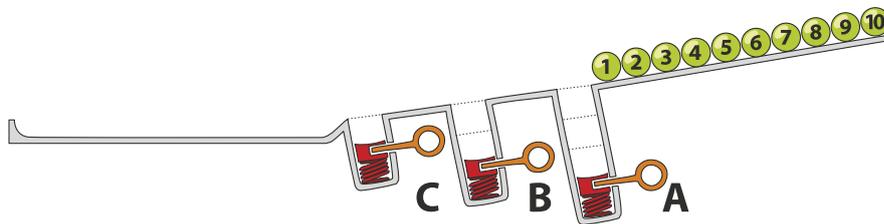
— https://fr.wikipedia.org/wiki/Étude_du_petit_monde



6. 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 ?

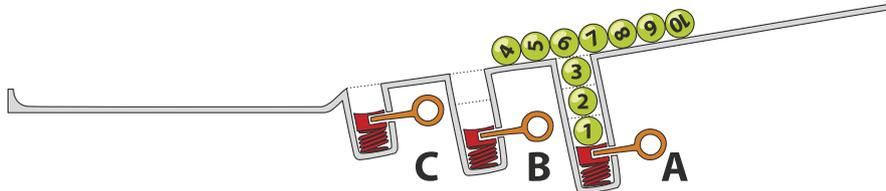
- A) B) C) D)



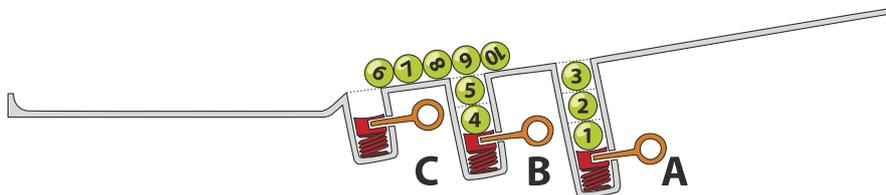
Solution

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

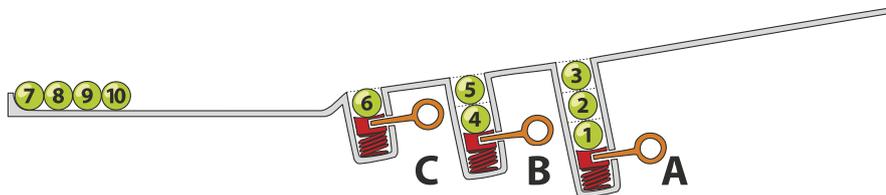
Les billes 1, 2 et 3 tombent dans le trou A, les billes 4 à 10 passent le trou A (qui, lui, contient les billes 1 à 3).



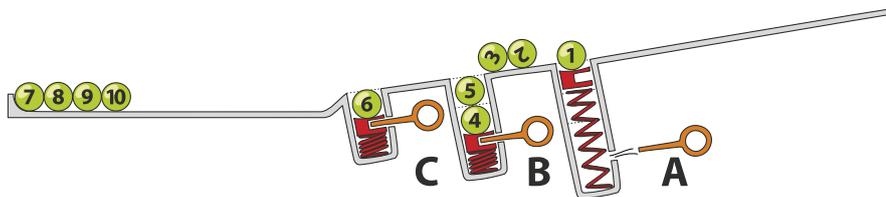
Ensuite, les billes 4 et 5 tombent dans le trou B et les billes 6 à 10 passent le trou B (qui, lui, contient les billes 4 à 5).



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).

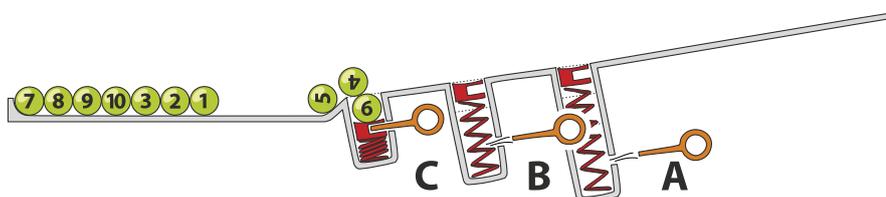


Quand les billes 7 à 10 ont passé les trous, le ressort du trou A libère les billes 3, 2, 1. Celles-ci se retrouvent donc de nouveau sur la rampe et roulent en direction de la fin de la rampe.



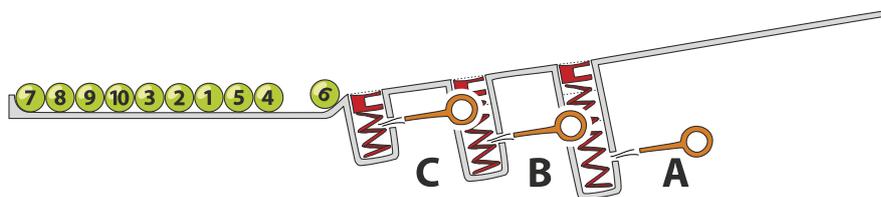
L'ordre de la séquence est donc 7, 8, 9, 10, 3, 2, 1.

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

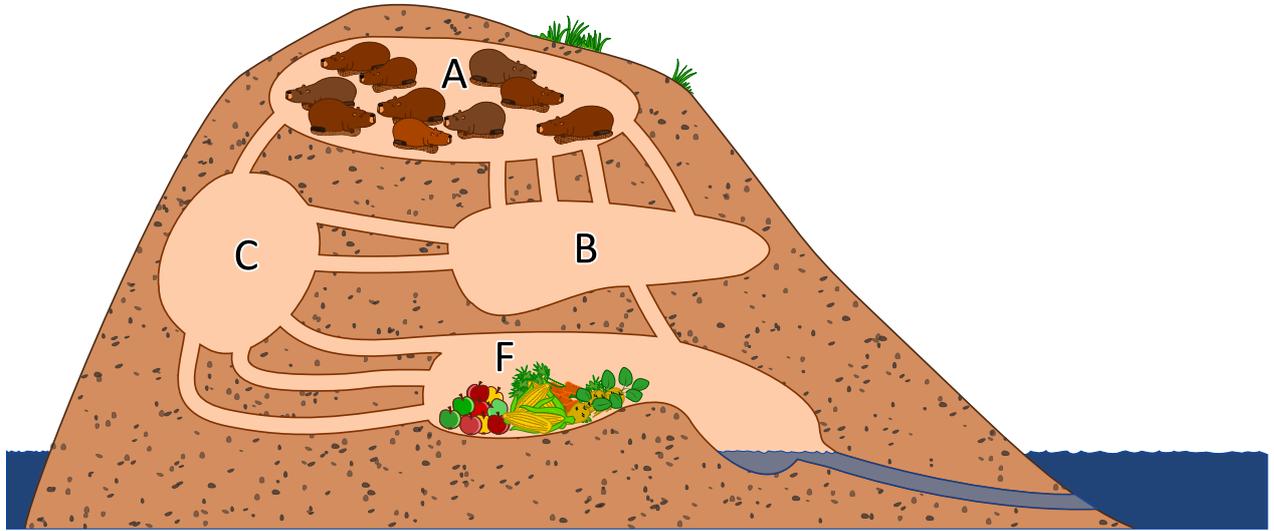
Stack (en français pile), LIFO

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





7. 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

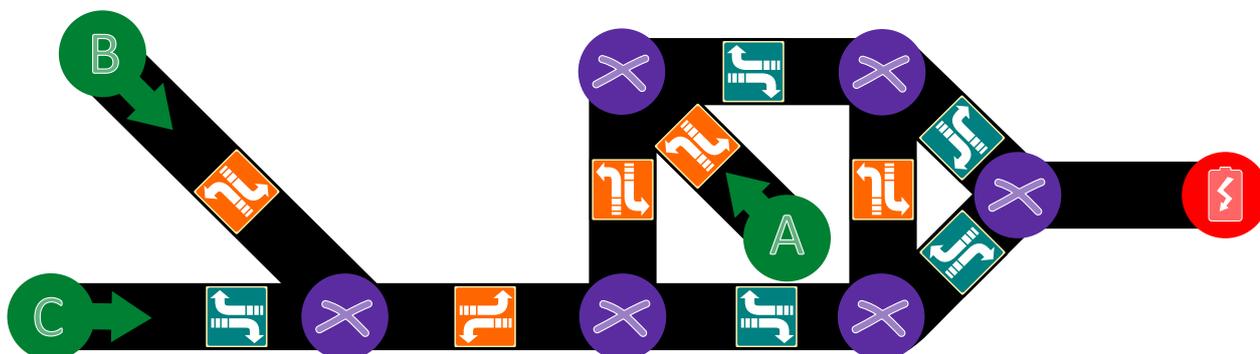
- https://fr.wikipedia.org/wiki/Réseau_de_flot
- https://fr.wikipedia.org/wiki/Graphe_orienté
- https://fr.wikipedia.org/wiki/Algorithme_de_Ford-Fulkerson





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



9. 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



10. 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



11. 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>





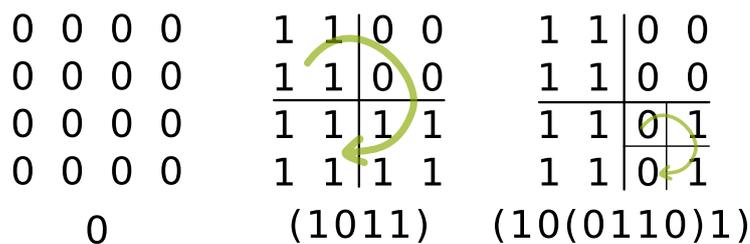
12. La méthode Quadtree

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.

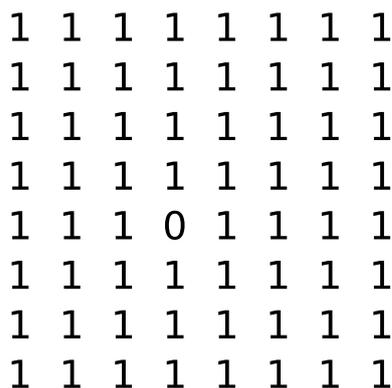
Beaucoup d'images peuvent toutefois être représentées par moins de caractères si on utilise la méthode *des quarts*. Pour ce faire, les caractères binaires sont disposés sur une grille carrée. Puis, on applique la méthode *des quarts* comme suit :

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

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

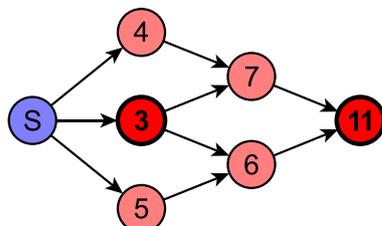
— <https://fr.wikipedia.org/wiki/Quadtree>





13. 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 ?

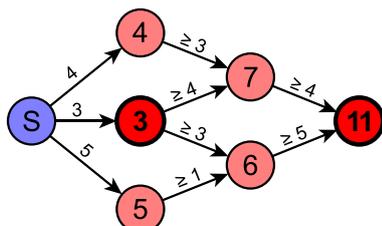
- A) La longueur du parcours minimal entre ces deux carrefours est exactement de 8.
- B) La longueur du parcours minimal entre ces deux carrefours est de 8 ou moins de 8.
- C) La longueur du parcours minimal entre ces deux carrefours est de 8 ou plus de 8.
- D) On ne peut rien dire de précis sur la longueur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



14. 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



15. 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
 Wilfried Baumann
 Bartosz Bieganski
 Daphne Blokhuis
 Eugenio Bravo
 Carmen Bruni
 Marios Choudary
 Zsófia Csepregi-Horváth
 Valentina Dagienė
 Christian Datzko
 Susanne Datzko
 Janez Demšar
 Olivier Ens
 Hanspeter Erni
 Michael Fellows
 Gerald Futschek

 Yasemin Gülbahar
 Martin Guggisberg
 Urs Hauser
 Juraj Hromkovič
 Ungyeol Jung
 Filiz Kalelioğlu
 Dong Yoon Kim
 Vaidotas Kinčius
 Ivana Kosírová
 Regula Lacher
 Milan Lukić
 Dario Malchiodi
 Dimitris Mavrovouniotis
 Henry Ong
 Wolfgang Pohl
 Ilya Posov

 Sergei Pozdniakov
 J.P. Pretti
 Frances Rosamond
 Kirsten Schlüter
 Victor Schmidt
 Eljakim Schrijvers
 Masood Seddighin
 Taras Shpot
 Seiichi Tani
 Jiří Vaníček
 Troy Vasiga
 Nicolette Venn
 Michael Weigend
 Hongjin Yeh



B. Sponsoring : Concours 2017

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

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

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

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

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

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

 <http://www.verkehrshaus.ch/>
i-factory (Musée des transports, Lucerne)

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

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

 <http://www.presentex.ch/>
PRESENTEX
Das Geschenk - die gute Werbung



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.