

SOINDEX?



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA



HEILBRONN → H416
4 6

KANT → K530
5 3

Exercices et solutions 2018

Années scolaires 9/10



LISSAJOUS → L222
2 2



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

CASTORO → C236
3 6 2

LAOYD → L300
3 0

Éditeurs :

Gabriel Parriaux, Jean-Philippe Pellet, Elsa Pellet, Julien Ragot, Christian Datzko, Susanne Datzko, Hanspeter Erni

BIBER → B160
6 1

GAUSS → G200
2 0

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

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischerverein für informatik in d
erausbildung // société suisse pour l'infor
matique dans l'enseignement // società sviz
zera per l'informatica nell'insegnamento



EULER → E460
6 4

CASTOR → C236
3 6 2





Ont collaboré au Castor Informatique 2018

Andrea Adamoli, Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Martin Guggisberg, Carla Monaco, Gabriel Parriaux, Elsa Pellet, Jean-Philippe Pellet, Julien Ragot, Beat Trachler.

Nous adressons nos remerciements à :

Juraj Hromkovič, Urs Hauser, Regula Lacher, Jacqueline Staub : ETHZ

Andrea Maria Schmid, Doris Reck : PH Luzern

Gabriel Thullen : Collège des Colombières

Valentina Dagienė : Bebras.org

Hans-Werner Hein, Ulrich Kiesmüller, Wolfgang Pohl, Kirsten Schlüter, Michael Weigend : Bundesweite Informatikwettbewerbe (BWINF), Allemagne

Chris Roffey : University of Oxford, Royaume-Uni

Anna Morpurgo, Violetta Lonati, Mattia Monga : ALaDDIn, Università degli Studi di Milano, Italie

Gerald Futschek, Wilfried Baumann : Oesterreichische Computer Gesellschaft, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Eljakim Schrijvers, Daphne Blokhuis, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers : Eljakim Information Technology bv, Pays-Bas

Roman Hartmann : hartmannGestaltung (Flyer Castor Informatique Suisse)

Christoph Frei : Chragokyberneticks (Logo Castor Informatique Suisse)

Andrea Adamoli (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 Elsa Pellet et la version italienne par Andrea Adamoli.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Le Castor Informatique 2018 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

Tous les liens ont été vérifiés le 1^{er} novembre 2018. Ce cahier d'exercice a été produit le 16 novembre 2018 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. 39.



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 » (<https://www.bebas.org/>), initié en Lituanie.

Le concours a été organisé pour la première fois en Suisse en 2010. Le Petit Castor (5^e et 6^e HarmoS / Castor 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 2018 a été fait pour cinq tranches d'âge, basées sur les années scolaires :

- 5^e et 6^e HarmoS / Castor 3 et 4 (Petit Castor)
- 7^e et 8^e HarmoS / Castor 5 et 6
- 9^e et 10^e HarmoS / Castor 7 et 8
- 11^e et 12^e HarmoS / Castor 9 et 10
- 13^e à 15^e HarmoS / Castor 11 à 13

Les élèves des 5^e et 6^e années HarmoS, aussi référencées comme années Castor 3 et 4, avaient 9 exercices à résoudre : 3 faciles, 3 moyens, 3 difficiles. Les élèves des 7^e et 8^e années HarmoS / Castor 5 et 6 avaient, quant à eux, 12 exercices à résoudre (4 de chaque niveau de difficulté). Finalement, chaque autre tranche d'âge devait résoudre 15 exercices (5 de chaque niveau de difficulté).

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

Chaque participant-e obtenait initialement 45 points (ou 27 pour la tranche d'âge « Petit Castor », et 36 pour les 7^e et 8^e années HarmoS / Castor 5 et 6).

Le nombre de points maximal était ainsi de 180 (ou 108 pour la tranche d'âge « Petit Castor », et 144 pour les 7^e et 8^e années HarmoS / Castor 5 et 6). Le nombre de points minimal était zéro.

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



Gabriel Parriaux

<https://www.castor-informatique.ch/fr/kontaktieren/>

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


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




Table des matières

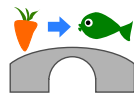
Ont collaboré au Castor Informatique 2018	i
Préambule	ii
1. Chutes d'eau	1
2. L'étang des castors	3
3. Compétition des castors	5
4. Maison numéro 29	7
5. Un extraterrestre !	9
6. Voisins	11
7. Jeu vidéo	15
8. Tournée des castors	17
9. Deux castors au travail	21
10. Marelle	23
11. Cadeaux	25
12. Rangées et colonnes	27
13. Classement de livres	31
14. Soundex	35
15. Trois amis	37
A. Auteurs des exercices	39
B. Sponsoring : Concours 2018	40
C. Offres ultérieures	42




1. Chutes d'eau

 Katja est au sommet d'une montagne. Cette montagne a trois chutes d'eau qui se rejoignent dans une rivière en bas de la vallée.

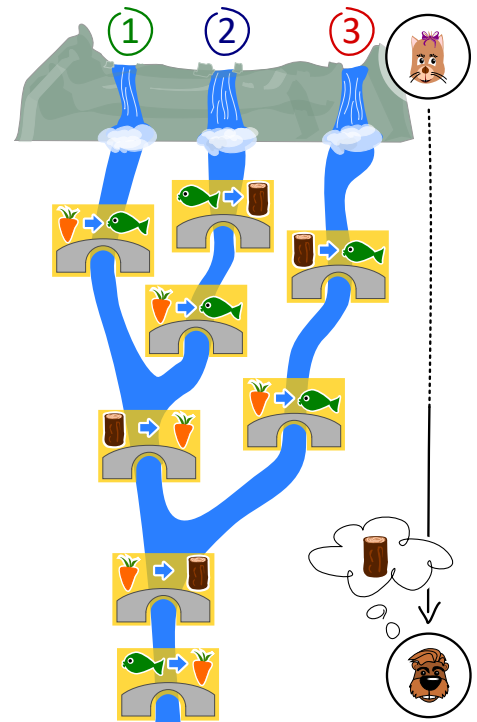
Katja peut lancer un poisson ou une carotte dans l'une des trois chutes d'eau. Les cours d'eau sont enjambés par plusieurs ponts sous lesquels vivent des trolls. Les trolls remplacent les objets passant sous les ponts par d'autres objets.



Par exemple, si une carotte passe sous un pont comme celui ci-dessus, les trolls la remplacent par un poisson.

 Justus est au bord de la rivière en bas de la vallée. *Justus a besoin de bois. Quel objet Katja doit-elle lancer dans quelle chute d'eau afin que Justus reçoive du bois ?*

- A) Elle lance un poisson 🐟 dans la chute d'eau numéro 1.
- B) Elle lance un poisson 🐟 dans la chute d'eau numéro 2.
- C) Elle lance une carotte 🥕 dans la chute d'eau numéro 2.
- D) Elle lance une carotte 🥕 dans la chute d'eau numéro 3.





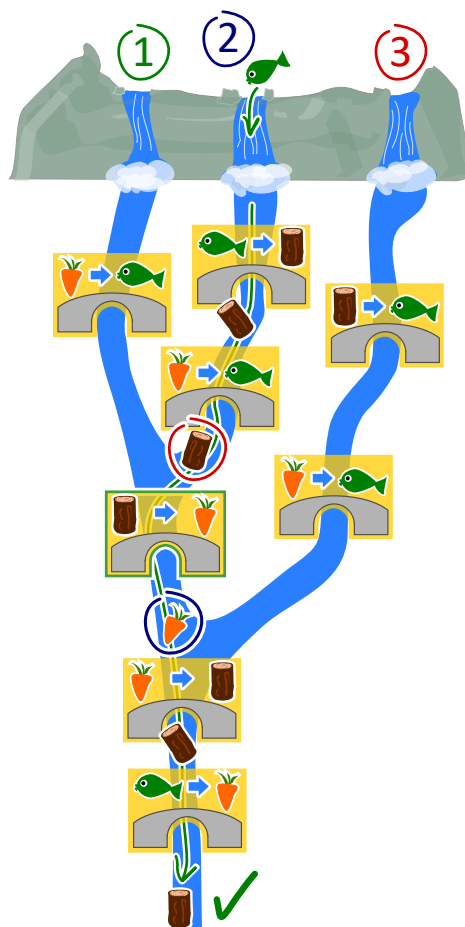
Solution

La bonne réponse est B). Elle lance un poisson 🐟 dans la chute d'eau numéro 2.

Voilà ce qu'il se passe lors des différentes solutions proposées :

- A) Un poisson lancé dans la chute d'eau numéro 1 ne sera remplacé que sous le dernier pont. Justus reçoit donc une carotte.
- B) Un poisson lancé dans la chute d'eau numéro 2 est remplacé par du bois, qui est remplacé par une carotte, elle-même ensuite remplacée par du bois. Justus reçoit donc du bois. C'est la bonne solution.
- C) Une carotte lancée dans la chute d'eau numéro 2 est remplacée par un poisson, qui est ensuite remplacé par une carotte. Justus reçoit donc une carotte.
- D) Une carotte lancée dans la chute d'eau numéro 3 est remplacée par un poisson, qui est ensuite remplacé par une carotte. Justus reçoit donc une carotte.

Une autre manière de résoudre cet exercice consiste à commencer par la fin : Pour obtenir du bois en bas de la montagne, l'objet flottant doit être une carotte quand il passe sous l'avant-dernier pont. La seule possibilité d'avoir une carotte à cette étape 🥕 est que du bois passe sous le pont commun des chutes numéro 1 et 2 (et pas 3) 🪵. La seule possibilité d'avoir du bois à cet endroit est de lancer un poisson dans la chute d'eau numéro 2.



C'est de l'informatique !

On peut se représenter un ordinateur comme une machine qui lit des données, les traite, puis écrit des données sortantes. Mais comment l'ordinateur « sait »-il comment traiter les données ? Il a reçu des ordres concernant les tâches à accomplir. Ceci se fait en écrivant des programmes.

Il existe beaucoup de langages de programmation qui fonctionnent selon différents paradigmes. La programmation fonctionnelle est un tel paradigme. Ce style de programmation est lui-même comme un petit ordinateur : il est composé de beaucoup de fonctions (ou routines) qui traitent des données et retournent des valeurs sortantes. Les ponts de cet exercice sont comme de petites fonctions, et le système complet comme un programme écrit dans un langage de programmation fonctionnel.

Mots clés et sites web

Paradigme de programmation, programmation fonctionnelle, fonctions et paramètres

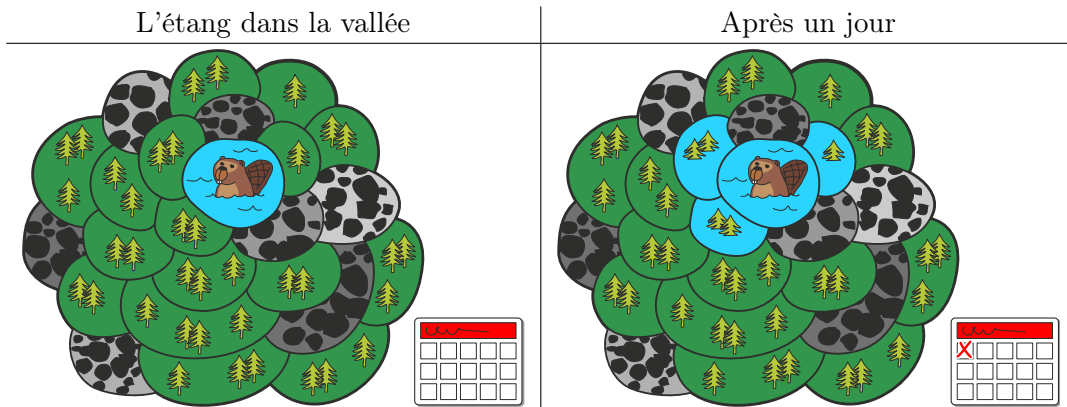
- [https://fr.wikipedia.org/wiki/Test_\(informatique\)](https://fr.wikipedia.org/wiki/Test_(informatique))
- https://fr.wikipedia.org/wiki/Boîte_blanche
- https://fr.wikipedia.org/wiki/Test_de_la_boîte_noire
- [https://fr.wikipedia.org/wiki/Paradigme_\(programmation\)](https://fr.wikipedia.org/wiki/Paradigme_(programmation))
- https://fr.wikipedia.org/wiki/Programmation_fonctionnelle
- [https://fr.wikipedia.org/wiki/Routine_\(informatique\)](https://fr.wikipedia.org/wiki/Routine_(informatique))



2. L'étang des castors

Il y a un petit étang dans une vallée. Il est entouré de parcelles de terrain forestier ou rocailloux. Plusieurs castors vivent dans l'étang.

Il vient un jour où les castors trouvent l'étang trop petit et décident d'inonder des parcelles de forêt. Chaque jour, ils inondent toutes les parcelles de forêt partageant une bordure avec une parcelle déjà inondée. Trois parcelles de forêt sont inondées le premier jour.



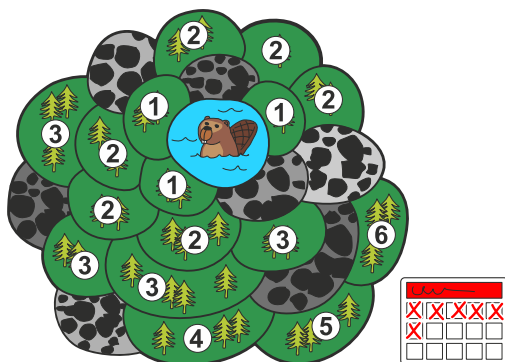
Après combien de jours en tout (y compris le premier jour représenté plus haut) les parcelles forestières sont-elles toutes inondées ?



Solution

Toutes les parcelles de forêt sont inondées en six jours.

L'image ci-dessous montre au combienième jour chaque parcelle a été inondée. Les parcelles voisines de l'étang sont inondées après le premier jour et donc marquées du chiffre 1. Les parcelles voisines de ces dernières sont marquées du chiffre 2 ; elles sont inondées après le deuxième jour, et ainsi de suite. La dernière parcelle est marquée du chiffre six et est inondée le sixième jour – toutes les parcelles forestières sont donc inondées à ce moment-là.



C'est de l'informatique !

Dans cet exercice, les castors inondent un espace forestier connexe qui est composé, en plus de l'étang, de parcelles séparées. L'espace est connexe car l'on peut atteindre chaque parcelle de forêt en passant par d'autres parcelles sans sortir de l'espace forestier.

Il existe également en dehors de la vallée de l'étang des castors des espaces connexes devant être inondés. Une zone de couleur unie sur une image n'est finalement rien d'autre qu'un espace connexe de pixels de la même couleur. Un groupe d'adolescents, dans lequel chacun est relié à chacun par des liens d'amitiés directs ou par l'ami d'un ami d'un ami, est également un « espace connexe », si l'on considère un lien d'amitié entre deux personnes comme du voisinage.

En informatique, il y a des méthodes permettant de découvrir et d'investiguer les espaces connexes, comme les algorithmes de parcours en largeur ou en profondeur. Ces méthodes permettent par exemple de changer la couleur d'une zone sur une image ou de découvrir des groupements sur les réseaux sociaux.

Mots clés et sites web

Algorithme à front d'onde, algorithme de parcours en largeur

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

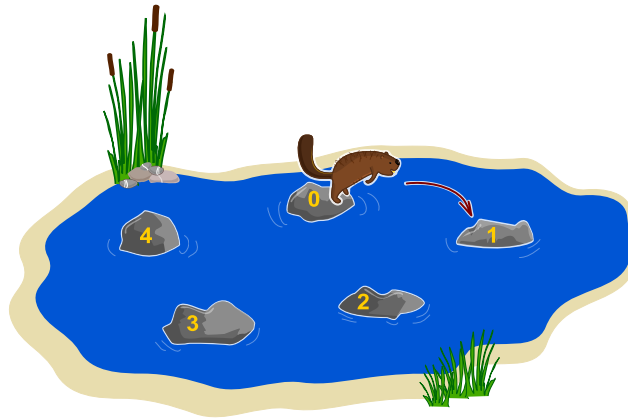


3. Compétition des castors

Plusieurs castors suivent un entraînement intensif en préparation à la compétition annuelle des castors. L'entraînement du jour consiste en un parcours de saut de pierre en pierre, dans le sens des aiguilles d'une montre, comme indiqué par la flèche. Si le castor saute 8 fois, il termine son parcours sur la pierre numéro 3 :

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3.$

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3.$



Le castor le plus fort a sauté 129 fois aujourd'hui. Sur quelle pierre se trouvait-il lorsqu'il s'est arrêté ?



Solution

5 sauts amènent le castor sur la pierre de laquelle il est parti. Ces 5 sauts constituent un tour. Pour déterminer où le castor arrive après 129 sauts, nous devons déterminer combien de tours il fait, ainsi que combien de sauts il fait après le dernier tour complet. Ce sont dans ce cas-là $129 = 25 \cdot 5 + 4$ sauts (25 tours et 4 sauts), ce qui veut dire que le castor termine son parcours sur la même pierre que s'il n'avait sauté que 4 fois, donc sur la pierre numéro 4.

C'est de l'informatique !

Tu as peut-être déjà rencontré une opération semblable en cours de mathématique ; il s'agit d'une *division avec reste*, appelée aussi division euclidienne ou division posée.

Dans cet exercice, il s'agit d'effectuer la division euclidienne de $129 : 5 = 25$ reste 4. En informatique, le calcul du reste est fréquemment utilisé et a donc un nom spécifique : le modulo. Les signes « % » ou « mod » sont habituellement utilisés comme opérateurs, on peut donc écrire $129 \% 5 = 4$.

Cet opérateur est par exemple utilisé dans des boucles (comme le castor qui fait des tours en sautant) lorsque que des variables débordent et dans un procédé de cryptographie très répandu appelé « chiffrement RSA ».

Mots clés et sites web

Modulo-Operation

- [https://fr.wikipedia.org/wiki/Modulo_\(opération\)](https://fr.wikipedia.org/wiki/Modulo_(opération))
- https://fr.wikipedia.org/wiki/Division_posée
- https://fr.wikipedia.org/wiki/Division_euclidienne
- https://fr.wikipedia.org/wiki/Chiffrement_RSA



4. Maison numéro 29

Milo fait un stage dans un lotissement de maisons de vacances. Aujourd'hui, il doit fixer des plaques numérotées aux maisons de vacances. Certaines maisons sont déjà numérotées. Il commence par la maison numéro 50. Depuis là, il doit :

- aller à gauche si le nouveau numéro est plus petit que celui de la maison devant laquelle il se trouve,
- aller à droite si le nouveau numéro est plus grand que celui de la maison devant laquelle il se trouve,
- fixer la plaque numérotée à la maison devant laquelle il se trouve si celle-ci n'est pas encore numérotée.

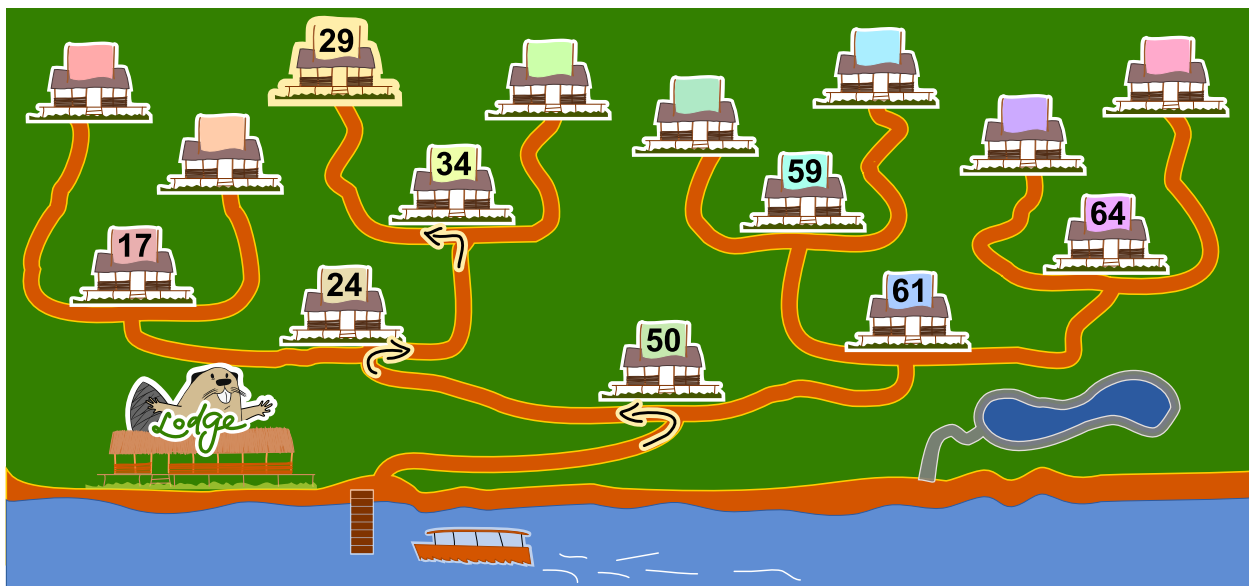


A quelle maison de vacances Milo doit-il fixer le numéro 29 ?



Solution

La maison de vacances correcte est la troisième depuis la gauche :



Le nouveau numéro 29 est plus petit que le numéro de la maison 50, donc Milo doit commencer par aller à gauche. Il arrive à la maison 24, dont le numéro est plus petit que 29, et doit donc aller à droite. Le numéro 29 est plus petit que celui de la maison 34 devant laquelle il arrive, donc il va à gauche. La maison suivante n'a pas encore de numéro, donc il y fixe la plaque numéro 29.

C'est de l'informatique !

La numérotation des maisons de vacances correspond à un *arbre binaire de recherche*, une structure de données souvent utilisée en informatique. Un arbre binaire de recherche permet de retrouver rapidement des données enregistrées.

Un arbre binaire de recherche est construit de telle façon qu'un « *élément* » est enregistré à chaque intersection (« *nœuds* »). Après chaque intersection, il y a au maximum deux chemins (« *arêtes* ») menant à d'autres intersections. Lors de l'enregistrement de nouvelles données, le chemin de gauche (par exemple) est toujours choisi lorsque le nouvel élément a une valeur plus petite que l'élément situé à l'intersection, et sinon le chemin de droite est suivi. Le nouvel élément est enregistré à la première intersection de libre.

Lorsque l'on cherche un élément précis, on peut ainsi facilement savoir quel chemin suivre à chaque intersection. Si l'arbre binaire de recherche est « *équilibré* » (appelé alors un *arbre AVL*), chaque étape de la recherche réduit de moitié le nombre d'intersections devant encore être au maximum parcourues avant de trouver l'élément recherché. Cela veut dire qu'un arbre de 1000 éléments peut être exploré en seulement 10 étapes, un arbre de 1'000'000 éléments en 20 étapes et un arbre de 1'000'000'000 éléments en 30 étapes (donc, pour n éléments, $\log_2(n)$ étapes).

Mots clés et sites web

Arbre binaire de recherche, arbre AVL

- https://fr.wikipedia.org/wiki/Arbre_binaire_de_recherche
- https://fr.wikipedia.org/wiki/Arbre_AVL

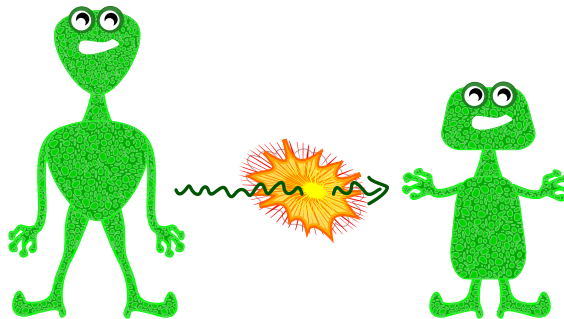


5. Un extraterrestre !

Un extraterrestre possède une tête, un tronc, deux bras et deux jambes. Cet extraterrestre peut être modifié par les instructions suivantes ; chaque partie du corps peut être modifiée plusieurs fois.

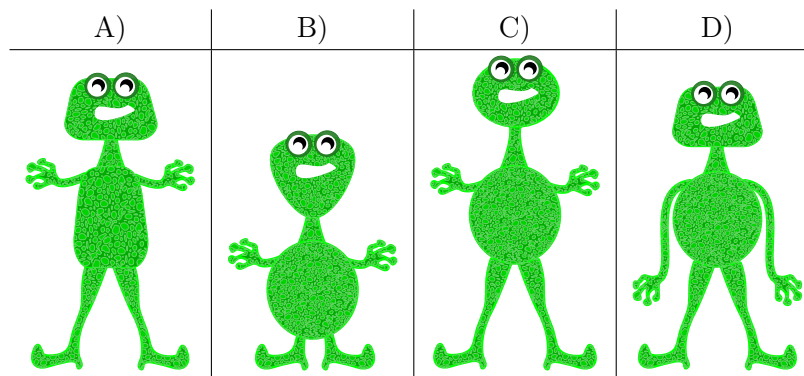
$Te(e)$	La tête devient elliptique.	
$Te(4)$	La tête devient rectangulaire.	
$Te(3)$	La tête devient triangulaire.	
$Tr(e)$	Le tronc devient elliptique.	
$Tr(4)$	Le tronc est rectangulaire.	
$Tr(3)$	Le tronc devient triangulaire.	
$B(+)$	Les bras deviennent longs.	
$B(-)$	Les bras deviennent courts.	
$J(+)$	Les jambes deviennent longues.	
$J(-)$	Les jambes deviennent courtes.	

Les instructions sont effectuées les unes après les autres de gauche à droite. Par exemple, les instructions $Te(e)$, $Tr(4)$, $Te(4)$, $B(-)$, $J(-)$ résultent en l'extraterrestre suivant :



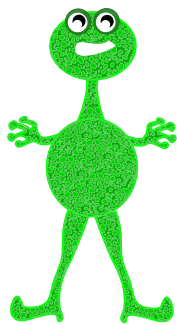
Quelle est l'apparence de l'extraterrestre après les instructions suivantes ?

$Te(3)$, $J(+)$, $Tr(3)$, $B(+)$, $Te(e)$, $B(-)$, $Tr(e)$





Solution



La bonne réponse est C)

La forme de chaque partie du corps est déterminée par la dernière instruction la concernant. Les instructions précédentes n'influencent pas l'apparence finale de l'extraterrestre, étant donné que l'instruction la plus récente remplace les instructions précédentes.

Le résultat des instructions ci-dessus est donc un extraterrestre avec un tronc elliptique ($Tr(e)$), des bras courts ($B(-)$), une tête elliptique ($Te(e)$) et de longues jambes ($J(+)$). Les autres extraterrestres proposés ont au moins deux caractéristiques différentes, et ne correspondent donc manifestement pas à la description.

C'est de l'informatique !

En informatique, lors de l'exécution d'un programme, les instructions sont traitées les unes après les autres. Les instructions actuelles peuvent donc remplacer ou modifier (« écraser ») les instructions les précédant.

Cela arrive souvent lorsque la valeur de variables doit être modifiée pendant l'exécution d'un programme. On peut imaginer que la forme de chaque partie du corps est enregistrée dans une variable. L'instruction « $Te(e)$ » enregistre alors la valeur « e » dans la variable nommée « Te ».

La notation « $Te(e)$ » est fonctionnelle. On appelle la fonction « $Te()$ » tout en lui donnant l'argument « e ». Cette notation est souvent utilisée, car la fonction « $Te()$ » peut aussi vérifier si l'argument, en l'occurrence « e », est valable. Si cela n'est pas nécessaire, où alors si la variable « Te » n'est utilisée que localement, on peut aussi directement écraser sa valeur en utilisant l'opérateur d'affectation « = ». On écrirait dans ce cas « $Te = e$ ».

Mots clés et sites web

Variable, séquence

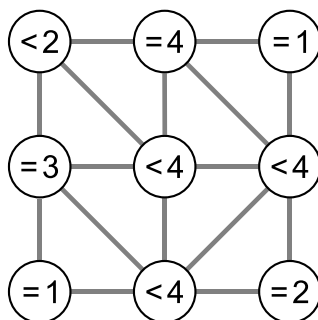
- [https://fr.wikipedia.org/wiki/Variable_\(informatique\)](https://fr.wikipedia.org/wiki/Variable_(informatique))
- https://fr.wikipedia.org/wiki/Programmation_structurée



6. Voisins

L'image ci-dessous montre neuf cercles partiellement connectés les uns aux autres. Une connexion entre deux cercles en fait des voisins. Les cercles peuvent être sélectionnés par un clic ; ils sont alors colorés en vert, alors que les cercles non-sélectionnés sont blancs.

Dans chaque cercle, une expression indique combien de cercles doivent être sélectionnés parmi les cercles voisins. Par exemple, le cercle portant l'expression « = 3 » doit avoir trois de ses quatre voisins sélectionnés, et les cercles portant l'expression « < 4 » peuvent en avoir au maximum trois qui sont sélectionnés.

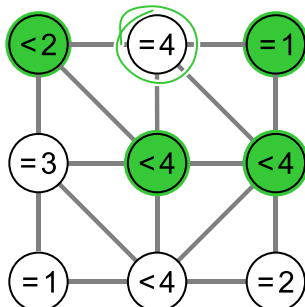


Sélectionne les cercles de manière à ce que toutes les conditions soient remplies.

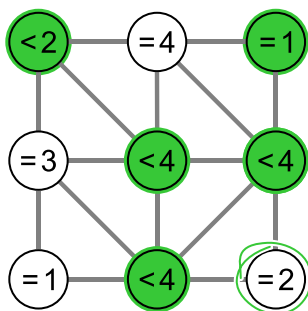


Solution

Le cercle central de la rangée du haut porte l'expression « = 4 » et a quatre voisins, ce qui fait que chacun de ses quatre voisins doit être sélectionné :



Similairement, le cercle en bas à droite porte l'expression « = 2 » et possède deux voisins, qui doivent donc être les deux sélectionnés (le cercle du centre à droite est déjà sélectionné) :



Toutes les conditions sont ainsi remplies.

Si l'on sélectionnait un cercle supplémentaire, certaines conditions seraient violées :

- Si le cercle « = 4 » au centre de la rangée du haut était sélectionné, la condition « = 1 » du cercle en haut à droite ne serait plus remplie.
- Si le cercle « = 3 » à gauche de la rangée centrale était sélectionné, la condition « < 2 » du cercle en haut à gauche ne serait plus remplie.
- Si le cercle « = 1 » en bas à gauche était sélectionné, la condition « = 3 » du cercle à gauche de la rangée centrale ne serait plus remplie.
- Si le cercle « = 2 » en bas à droite était sélectionné, la condition « < 4 » du cercle à droite de la rangée centrale ne serait plus remplie.

C'est de l'informatique !

De combien d'essais a-t-on besoin pour résoudre le problème ? Si l'on essaie simplement chaque solution possible, on a deux réglages différents pour chacun des neuf cercles, ce qui fait $2^9 = 512$ possibilités différentes. On appelle cette méthode *recherche exhaustive* ou *recherche par force brute*. Pour chacune de ces 512 possibilités, il faudrait vérifier si toutes les conditions sont remplies.

Dans ce cas-là, il est plus judicieux de procéder de manière logique et cohérente. On cherche d'abord les cercles ayant des conditions ne pouvant être remplies que de manière unique. Il s'agit par exemple des cercles ayant exactement n connexions, donc n voisins, et la condition « = n ». On peut continuer à partir de là de manière logique en regardant s'il reste des conditions que l'on ne peut remplir que d'une seule façon. On peut ainsi trouver la bonne solution à moindre effort. De manière générale, on peut trouver de cette manière qu'il n'existe pas de solution, ou trouver au moins une solution parmi



plusieurs. On parle de démarche analytique lorsque seule une solution prometteuse parmi toutes les possibilités est considérée « *heuristique* ».

Mots clés et sites web

Voisinage dans la théorie de graphes, approche logique

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





7. Jeu vidéo

Andrea a programmé un jeu vidéo à l'école. Les règles sont simples :

Le jeu se joue en plusieurs tours. Une feuille tombe lors de chaque tour. Le castor essaie d'attraper la feuille avant qu'elle ne touche le sol. Le castor gagne s'il attrape 15 feuilles avant que 4 feuilles ne touchent le sol.

La durée du jeu est égale au nombre de tours (et donc au nombre de feuilles tombées en tout).

Dans l'exemple suivant, le castor perd après 6 tours, car il a atteint le maximum de 4 feuilles touchant le sol. La durée du jeu dans cet exemple est de 6 tours.



Tour	Résultat	Score – nombre total de feuilles	
		Attrapées	Pas attrapées
1	Attrapée	1	0
2	Pas attrapée	1	1
3	Attrapée	2	1
4	Pas attrapée	2	2
5	Pas attrapée	2	3
6	Pas attrapée	2	4

Quelle est la durée maximale d'un jeu ?

- A) 4 tours
- B) 15 tours
- C) 18 tours
- D) 19 tours
- E) 20 tours
- F) La durée du jeu est illimitée.



Solution

Afin de déterminer la durée maximale d'un jeu, nous devons combiner toutes les situations lors desquelles le jeu continue. Pour ceci, nous combinons le nombre maximal de feuilles attrapées avant la fin du jeu (14 tours) avec le nombre maximal de feuilles touchant le sol avant la fin du jeu (3 tours). Au tour suivant, on peut soit attraper une 15^e feuille, soit en laisser tomber une 4^e. La durée maximale est donc $15 + 3 = 14 + 4 = 18$ tours et la bonne réponse est C).

La réponse A) 4 tours est la durée minimale du jeu si aucune feuille n'est attrapée.

La réponse B) 15 tours est la durée minimale du jeu si toutes les feuilles sont attrapées.

Les réponses D), E) et F) sont fausses, car le nombre maximal de feuilles attrapées ou pas attrapées est atteint avant.

C'est de l'informatique !

Lors de la programmation d'un jeu, les règles doivent être clairement définies. Les conséquences des règles doivent être bien comprises afin que le jeu permette de gagner et de perdre (le nombre de feuilles doit être suffisant) et que le jeu ne dure ni trop longtemps, ni pas assez.

Un jeu consistant en plusieurs tours est un processus, c'est-à-dire une suite d'opérations ordonnées. Les informaticiens sont des spécialistes de la modélisation et description de processus. Une des tâches principales est de déterminer tout ce qui peut se passer lors du déroulement d'un processus et combien de temps celui-ci peut durer.

Mots clés et sites web

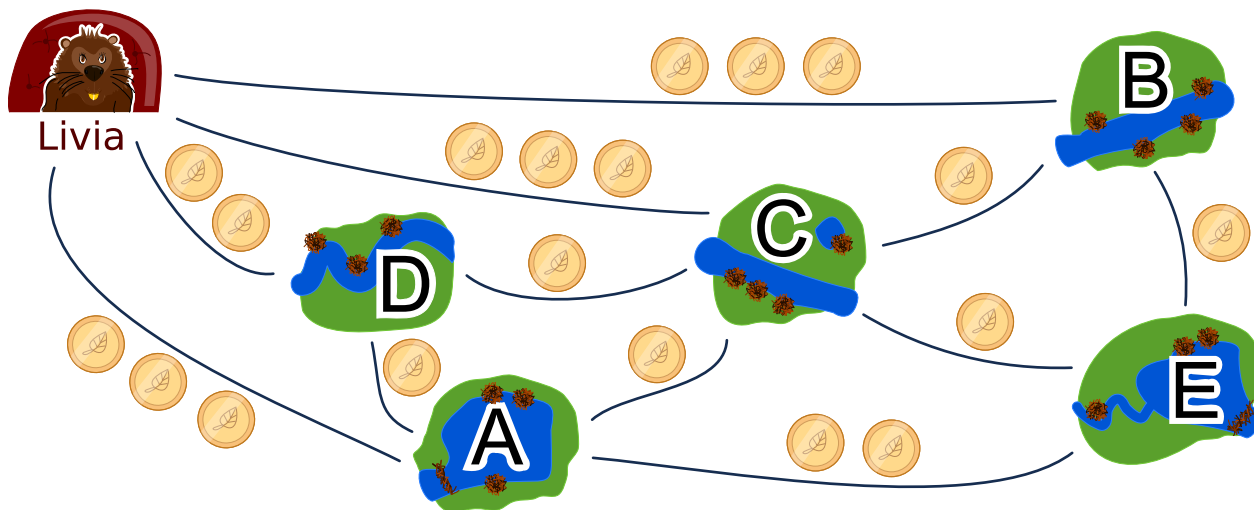
Analyse, vérification et validation de logiciel

- https://en.wikipedia.org/wiki/Software_verification
- https://en.wikipedia.org/wiki/Verification_and_validation



8. Tournée des castors

Livia aimerait rendre visite à chacun de ses amis dans les villages A, B, C, D et E en transports publics. Elle fait la tournée de tous ses amis lors d'un seul voyage, sans passer deux fois par le même village. Elle rentre chez elle à la fin de sa tournée de visites. Le prix de transport de chaque ligne est affiché ci-dessous.



Une des routes possible pour voir ses amis est :

départ → B → E → A → D → C → départ.

Cette route coûte $3 + 1 + 2 + 1 + 1 + 3 = 11$ francs castor.

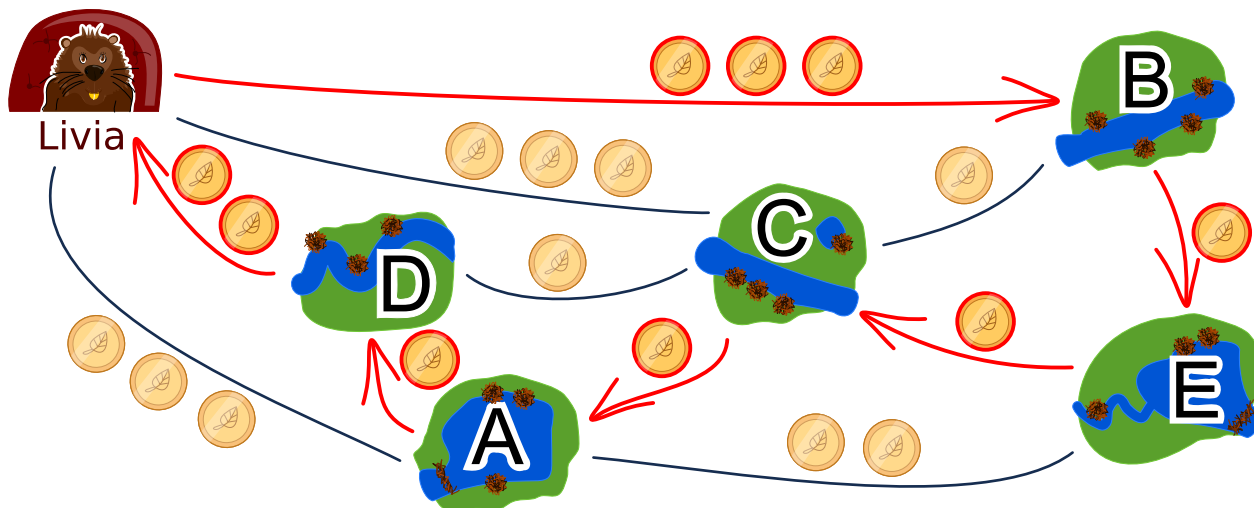
Dans quelle ordre Livia doit-elle rendre visite à ses amis ?



Solution

Il existe deux solutions optimales :

- départ → B → E → C → A → D → départ
- départ → D → A → C → E → B → départ



Les deux solutions sont semblables, mis à part le sens de trajet, et coûtent 9 francs castor. Il n’y a pas de meilleure solution. Depuis chez elle, Livia peut prendre une fois le chemin à 2 francs castor puis un chemin à 3 francs castor dans l’autre sens. Les quatre autres nœuds à visiter correspondent à quatre autres chemins coûtant au moins un franc castor chacun, ce qui fait déjà 9 francs castor au total.

Toutes les autres solutions sont plus chères :

- Coût de 10 francs castor : départ → A → D → C → E → B → départ
- Coût de 10 francs castor : départ → A → E → B → C → D → départ
- Coût de 10 francs castor : départ → B → C → E → A → D → départ
- Coût de 10 francs castor : départ → B → E → A → C → D → départ
- Coût de 10 francs castor : départ → B → E → C → D → A → départ
- Coût de 10 francs castor : départ → C → B → E → A → D → départ
- Coût de 10 francs castor : départ → D → A → E → B → C → départ
- Coût de 10 francs castor : départ → D → A → E → C → B → départ
- Coût de 10 francs castor : départ → D → C → A → E → B → départ
- Coût de 10 francs castor : départ → D → C → B → E → A → départ
- Coût de 11 francs castor : départ → B → E → A → D → C → départ
- Coût de 11 francs castor : départ → C → D → A → E → B → départ

Une des méthodes pour trouver le parcours le moins cher consiste à emprunter le chemin le moins cher, puis à chercher un solution à partir de là.

C’est de l’informatique !

La recherche de bonnes solutions, voire de solutions optimales, est l’un des problèmes fondamentaux de l’informatique. La description de notre problème d’optimisation peut être visualisée dans un diagramme ayant les amis comme nœuds et les chemins comme arêtes. La tâche consiste à passer par chacun des nœuds tout en minimisant la somme des poids des arêtes (le coût en francs castor). C’est un exercice semblable au célèbre problème du voyageur de commerce (Travelling Salesman Problem, TSP).



Ce type de problème est habituellement très difficile à résoudre de manière computationnelle. Afin d'éviter de devoir essayer chaque solution possible, on peut utiliser une bonne heuristique (un exemple d'heuristique est de commencer par le chemin le plus court) puis éliminer toutes les solutions qui sont moins bonnes. Dans ce cas, nous ne permettons qu'un passage par nœud. Si plusieurs passages par nœud étaient permis, le problème deviendrait plus complexe car il faudrait prendre beaucoup plus de possibilités en considération.

Mots clés et sites web

Optimisation, problème du voyageur de commerce

- https://fr.wikipedia.org/wiki/Problème_du_voyageur_de_commerce
- https://en.wikipedia.org/wiki/Optimization_problem
- [https://fr.wikipedia.org/wiki/Optimisation_\(mathématiques\)](https://fr.wikipedia.org/wiki/Optimisation_(mathématiques))

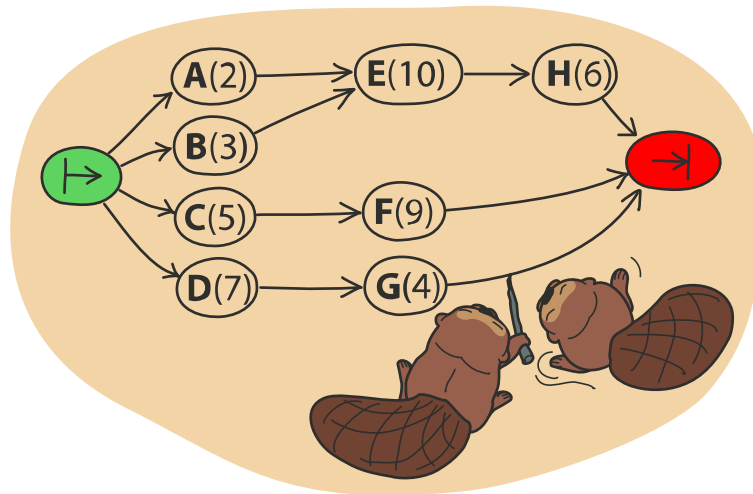




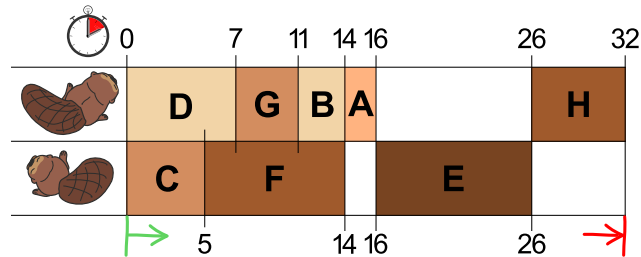
9. Deux castors au travail

Deux castors construisent un barrage et doivent pour cela réaliser huit tâches : abattre des arbres, enlever les branches des troncs, amener les troncs dans l'eau, et ainsi de suite. Chaque tâche est définie par une lettre (son nom) et un chiffre entre parenthèses qui donne le nombre d'heures de travail nécessaire à la réalisation de la tâche.

Certaines tâches ne peuvent être commencées que lorsque certaines autres sont terminées. Ce déroulement est représenté par des flèches dans le schéma ci-dessous. Les deux castors peuvent travailler en même temps à différentes tâches, mais ils ne peuvent pas travailler ensemble à la même tâche.



L'image ci-dessous montre un plan de travail possible pour les deux castors qui prévoit 32 heures de travail en tout, mais c'est possible de réaliser le barrage plus rapidement !



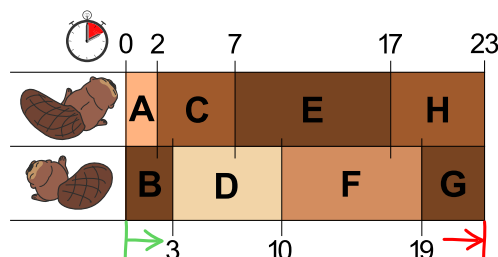
De combien de temps les castors ont-ils au minimum besoin pour construire le barrage ?



Solution

Au moins 23 heures sont nécessaires à la construction.

L'image dans la donnée montre un plan de travail possible. Dans celui-ci, le premier castor a une longue pause de 10 heures et le deuxième deux pauses de huit heures en tout. S'ils travaillaient sans pause, ils finiraient plus rapidement.



Si l'on veille à ce que les deux longues tâches E(10) et F(9) ne soient pas réalisées par le même castor, on trouve facilement un plan de travail qui prévoit 23 heures en tout. Ce n'est pas possible de construire le barrage plus rapidement, car les deux castors travaillent déjà sans interruption.

C'est de l'informatique !

Une possibilité pour trouver un des plans de travail les plus courts serait de suivre la règle suivante : « Choisis parmi les tâches à réaliser celle qui nécessite le plus d'heures de travail ». En informatique, on parle alors de stratégie *gloutonne*. On commence par réaliser les tâches qui nous font progresser le plus vers la solution finale du problème.

Les stratégies gloutonnes fonctionnent bien dans beaucoup de cas, mais parfois – comme dans cet exercice – elles ne sont pas adaptées. Cet exercice a été développé de manière à ce que la stratégie gloutonne ne fonctionne pas. Le fait de trouver de tels problèmes peu pratiques à résoudre est une tâche importante : en informatique théorique, par exemple, on analyse de manière ciblée la pire des situations (« worst case ») pour un programme afin de pouvoir mieux estimer le temps d'exécution d'un algorithme.

Il n'existe qu'une manière sûre de trouver la meilleure solution à ce problème, c'est d'essayer tous les plans de travail possibles qui respectent les règles données. Mais lors de grands projets, le nombre de possibilités peut être tellement grand que l'élaboration d'un plan de travail durerait trop longtemps. C'est dans ces cas-là qu'une stratégie gloutonne entre en jeu, car elle permet de trouver relativement rapidement une solution suffisamment bonne, même si ce n'est pas la solution optimale.

Mots clés et sites web

Ordonnancement, algorithme glouton

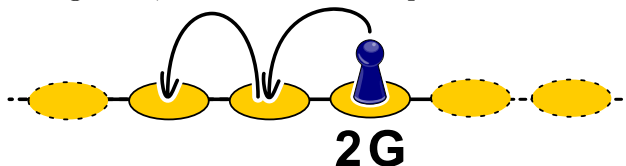
- https://fr.wikipedia.org/wiki/Ordonnancement_de_travaux_informatiques
- https://fr.wikipedia.org/wiki/Tri_topologique
- https://fr.wikipedia.org/wiki/Algorithme_glouton



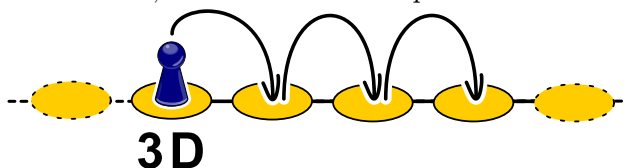
10. Marelle

Comme pour tout jeu de marelle, il s'agit ici de sauter de case en case en suivant certaines règles. Dans ce jeu-ci, une règle est associée à chaque case. Il y a trois sortes de règles :

- nG : sauter n cases vers la gauche, $2G$ veut donc dire qu'il faut sauter deux fois vers la gauche.

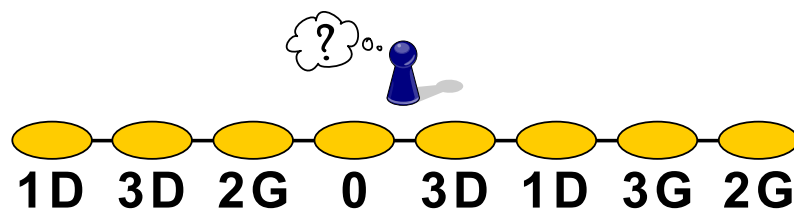


- nD : sauter n cases vers la droite, $3D$ veut donc dire qu'il faut sauter trois fois vers la droite.



- 0 : ne pas sauter plus loin.

De quelle case doit-on partir afin d'être passé une fois par chaque case durant le jeu ?

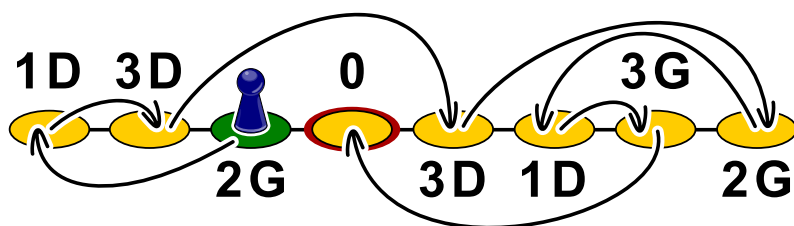




Solution

En partant de la troisième case depuis la gauche (« 2G »), on passe par chaque case avant d'avoir terminé le jeu.

On trouve la solution en cherchant la case depuis laquelle on peut atteindre la case « 0 ». Dans notre cas, il s'agit de la deuxième case depuis la droite (« 3G »). Celle-ci peut être atteinte depuis la troisième case depuis la droite (« 1D »), qui peut être atteinte depuis la case tout à droite (« 2G »). Celle-ci peut être atteinte depuis la quatrième case depuis la droite (« 3D »), après être passé par la deuxième case depuis la gauche (« 3D »), qui peut pour sa part être atteinte depuis la case tout à gauche (« 1D »), en partant de la dernière case par laquelle l'on n'est pas encore passé, soit la troisième depuis la gauche (« 2G »).



Le traçage du chemin par des flèches fait des cases un graphe orienté qu'il suffit de parcourir à l'envers depuis la case « 0 » pour arriver sur la case de départ, la troisième depuis la gauche.

C'est de l'informatique !

En informatique, la structure de données appelée « *liste chaînée* » fonctionne similairement aux cases de la marelle : dans la mémoire vive, les objets sont enregistrés dans des cellules contenant l'adresse de la cellule contenant l'objet suivant. Le système de gestion de la mémoire peut ainsi enregistrer un objet à n'importe quel endroit de la mémoire vive et n'a pas besoin de temps pour établir un espace de stockage connexe pour tous les objets. De son côté, le programmeur ne doit s'occuper de rien à part de réserver un espace de stockage de taille suffisante.

Mais que se passe-t-il lorsque des objets enregistrés ne sont plus utiles ? Alors qu'auparavant le programmeur devait s'occuper de libérer de l'espace de stockage (ce qui, malheureusement, créait souvent des problèmes, les programmes gaspillant de l'espace de stockage avant de se bloquer à cause d'un manque de mémoire), les langages de programmation modernes ont pour cela une sorte de service de ramassage des ordures, le « *ramasse-miettes* », qui vérifie régulièrement si les objets sont encore référencés (donc si d'autres objets possèdent leur adresse). Parfois, de grosses structures ne sont plus référencées, et il faut donc suivre les références jusqu'à l'adresse de départ comme dans l'exemple.

Mots clés et sites web

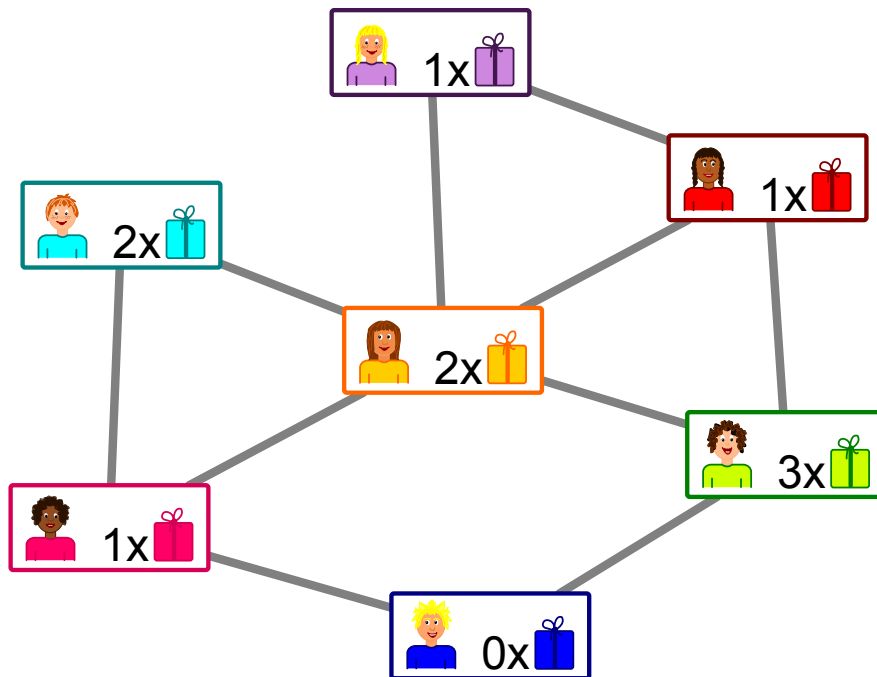
File, gestion de la mémoire, GOTO

- [https://fr.wikipedia.org/wiki/Ramasse-miettes_\(informatique\)](https://fr.wikipedia.org/wiki/Ramasse-miettes_(informatique))
- <https://en.wikipedia.org/wiki/St-connectivity>



11. Cadeaux

L'image suivante montre les liens d'amitiés entre les enfants habitant le même immeuble. Un trait reliant deux enfants signifie qu'ils sont amis.



Les habitants de l'immeuble organisent une fête avec des cadeaux pour les enfants. L'un des enfants de chaque paire d'amis doit offrir un cadeau à l'autre.



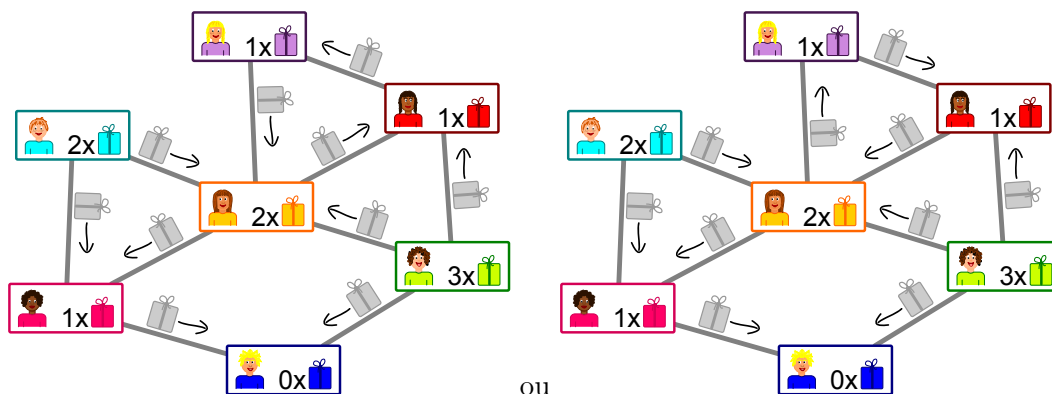
L'image montre le nombre de cadeaux que chaque enfant peut offrir : la fille en violet peut offrir un cadeau.

Tu n'as pas encore décidé qui offre le cadeau à qui pour chaque paire d'amis.



Solution

Il y a deux possibilités d'organiser la distribution de cadeaux sans qu'un enfant ne doive en offrir plus qu'il ne peut :



On commence par l'enfant tout en bas : il ne peut offrir aucun cadeau et va donc recevoir un cadeau de chacune de ses amies à gauche et à droite. Son amie à gauche ne pouvant offrir qu'un seul cadeau, elle va en recevoir de ses deux autres amis au-dessus et au centre. La direction des autres flèches est ainsi claire.

Le seul endroit où il reste un choix est la partie en haut à droite : les trois enfants peuvent s'offrir des cadeaux dans le sens des aiguilles d'une montre ou en sens inverse.

C'est de l'informatique !

Les liens d'amitié entre les enfants forment un réseau constitué de nœuds (les enfants) et d'arcs (les liens d'amitié). Les « réseaux sociaux » sont formés de manière similaire avec des millions d'utilisateurs. Ces systèmes possèdent un aspect pouvant toutefois les différencier fondamentalement : certains possèdent des « amitiés » réciproques pour lesquels les liens n'ont pas de direction, comme dans cet exercice. D'autres fonctionnent avec des abonnés (« follower » en anglais), ce qui fait que les liens ont une direction : par exemple, si tu suis une utilisatrice célèbre, cela ne veut pas dire qu'elle te suit également.

Dans cet exercice, il faut assigner une direction aux liens d'amitié pour une distribution de cadeaux. C'est un nouvel aspect du système, car chaque enfant a une capacité limitée d'offrir, ce qui met indirectement des limites au choix de la direction. Le but est qu'un cadeau soit offert dans chaque amitié sans dépasser la capacité d'offrir des enfants. Il existe des problèmes similaires en informatique : dans un réseau (comme par exemple les câbles constituant Internet), la capacité des connections est limitée. Cette capacité doit être utilisée de manière optimale tout en respectant les limites.

Le problème du débit maximum dans un réseau peut être résolu de manière efficace. Étant donné que la structure de notre exercice et celle du problème du débit maximum sont semblables, on peut appliquer la même méthode pour le résoudre. Ceci arrive souvent en informatique : un problème peut être transformé en un autre problème ayant la même structure qui a déjà été facilement résolu.

Mots clés et sites web

Débit dans un réseau, réduction de problème

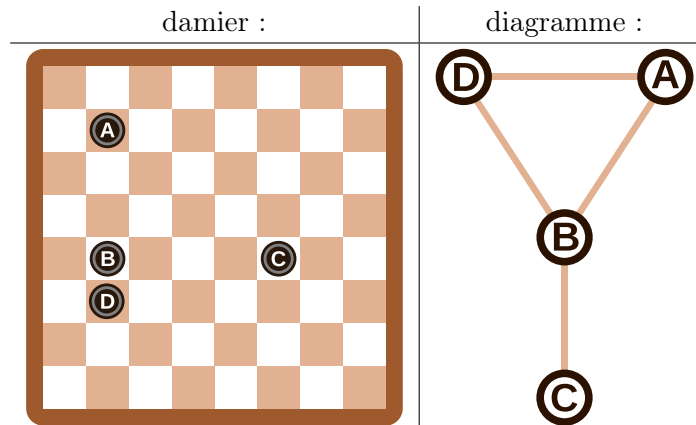
- https://fr.wikipedia.org/wiki/Problème_de_flot_maximum
- [https://fr.wikipedia.org/wiki/Réduction_\(complexité\)](https://fr.wikipedia.org/wiki/Réduction_(complexité))



12. Rangées et colonnes

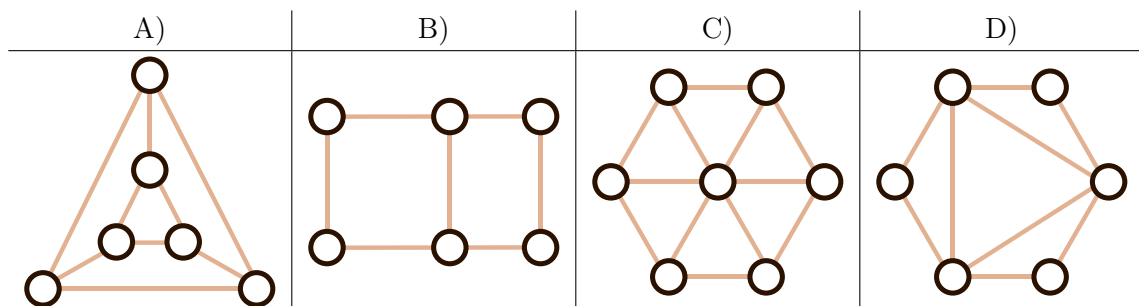
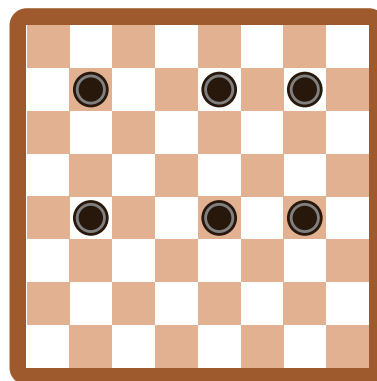
Le diagramme des palets de jeu montré à droite du damier a été construit de la manière suivante :

- Chaque palet est représenté par un cercle,
- deux palets sont reliés par une ligne s'ils se trouvent sur la même rangée ou colonne du damier.



Pour cet exemple, les palets sur le damier et les cercles du diagramme sont annotés d'une lettre afin de mettre leur relation en évidence.

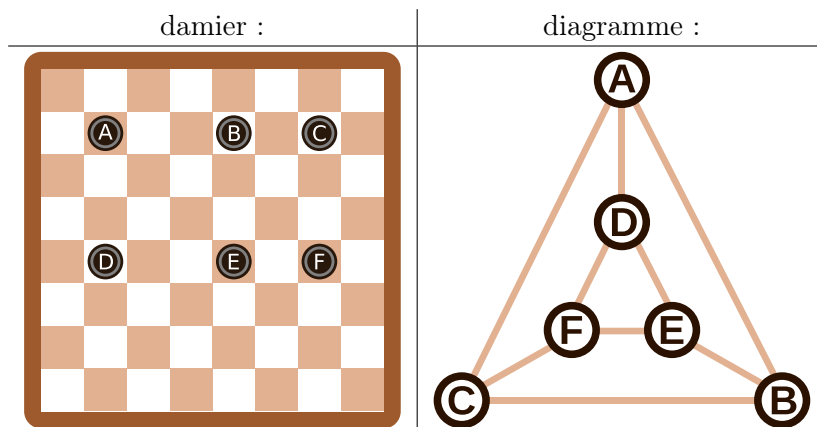
Quel diagramme correspond au damier à six palets suivant ?



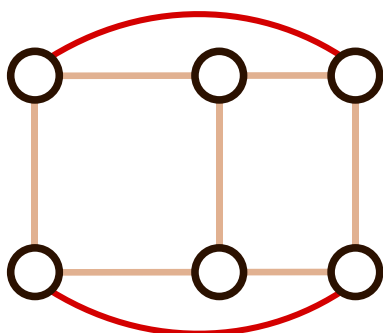


Solution

Le diagramme A) correspond au damier. On peut le vérifier à l'aide du graphique suivant sur lequel les palets et les cercles sont annotés :



Les diagrammes B), C) et D) peuvent être exclus de la manière suivante : chaque palet se trouve dans la même rangée que deux autres palets et dans la même colonne qu'un autre palet. Cela veut dire que chaque cercle dans le diagramme doit être relié à $2 + 1 = 3$ autres cercles, ce qui n'est le cas que sur le diagramme A). De plus, le diagramme C) comporte sept cercles, donc un de trop. Le diagramme B) est faux, même s'il ressemble au damier. Les quatre cercles extérieurs ne sont reliés qu'à deux autres cercles. Il faudrait ajouter deux lignes au diagramme pour le corriger :



C'est de l'informatique !

En informatique, de tels diagrammes sont souvent utilisés pour représenter les informations essentielles d'un problème. On appelle ces diagrammes des graphes. Les cercles sont appelés « nœuds » et les lignes « arêtes ».

L'important dans un graphe est de savoir quels nœuds sont reliés par des arêtes. L'arrangement des nœuds ou la forme des arêtes ne jouent aucun rôle. Le même graphe peut donc être représenté de différentes manières, comme nous l'avons vu plus haut : le graphe de la réponse A) et le dernier graphe de l'explication sont les deux des solutions correctes, deux représentations du même graphe. Les graphes sont une forme d'abstraction. Ils représentent l'essentiel d'un problème. Dans notre cas, on peut par exemple utiliser les graphes pour répondre à la question « Quel est le plus petit nombre de palets à enlever afin qu'il n'y ait jamais plus d'un palet par rangée et par colonne ? ». Une partie essentielle du travail d'informaticien consiste à trouver une bonne représentation du problème aidant à sa résolution.



Mots clés et sites web

Graphe

- [https://fr.wikipedia.org/wiki/Graphe_\(mathématiques_discrètes\)](https://fr.wikipedia.org/wiki/Graphe_(mathématiques_discrètes))
- https://fr.wikipedia.org/wiki/Théorie_des_graphes
- [https://fr.wikipedia.org/wiki/Graphe_\(type_abstrait\)](https://fr.wikipedia.org/wiki/Graphe_(type_abstrait))





13. Classement de livres

Trois castors sont assis chacun à une table avec deux livres. Ils veulent classer les livres voisins en échangeant leurs places. Chaque livre peut être déplacé au maximum une fois par tour. Les castors travaillent ensemble à chaque tour.

Il existe deux sortes de tours qui sont toujours effectués l'un après l'autre :

- A. Chaque castor peut (mais ne doit pas) inverser les deux livres sur sa table (exemple A).
- B. Chaque livre peut (mais ne doit pas) être échangé avec le livre le plus proche sur une table voisine (exemple B).

Au départ, les livres sont placés comme suit :



Lors du premier tour, chaque castor inverse les deux livres sur sa table.

Quel est le nombre de tours minimal nécessaire au classement des livres par ordre croissant, c'est-à-dire dans l'ordre 1, 2, 3, 4, 5, 6 ?

- A) trois tours
- B) quatre tours
- C) cinq tours
- D) six tours

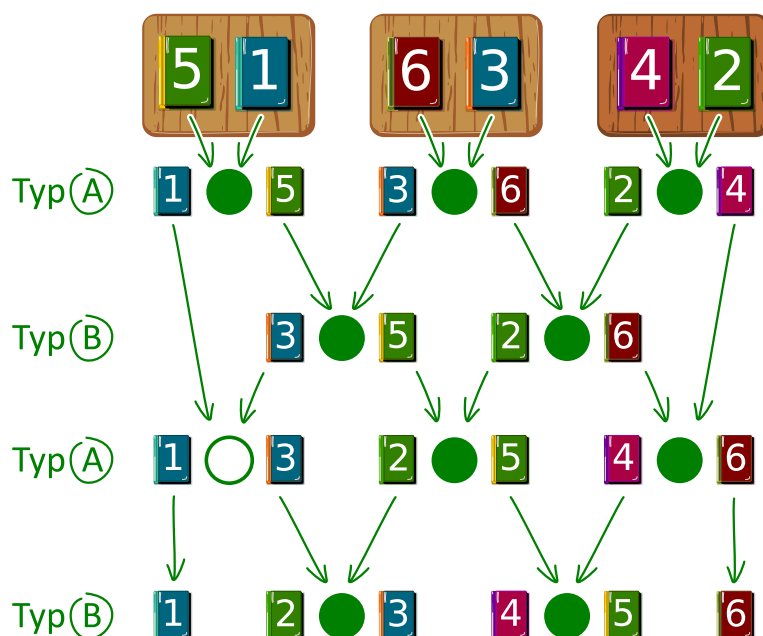


Solution

La bonne réponse est B).

L'illustration montre de quelle manière les livres peuvent être classés en échangeant leur place. Les castors utilisent une stratégie « gloutonne ». Cela veut dire qu'ils essaient de se rapprocher de la solution à chaque étape : ils comparent les livres voisins pouvant être échangés durant le tour en cours, et ne les échangent que s'ils ne sont pas encore dans le bon ordre (donc si le livre de gauche porte un numéro plus grand que celui de droite) ; s'ils sont déjà classés, ils ne font rien.

Lors du premier tour (sorte A), les deux livres sur chaque table sont échangés. Lors du deuxième tour (sorte B), les livres voisins situés sur les tables adjacentes sont échangés, lors du troisième tour (sorte A), seuls les livres sur les deux tables les plus à droites sont échangés. Finalement, lors du quatrième tour (sorte B), les quatre livres voisins situés sur des tables adjacentes sont inversés. Les livres sont maintenant classés. Ce n'est pas possible d'arriver à la solution en moins de tours, car le livre 5, par exemple, doit être déplacé de quatre positions en quatre tours pour arriver à la cinquième place.



C'est de l'informatique !

Le classement, ou tri, effectué dans cet exercice est un exemple d'algorithme parallèle. Plusieurs acteurs travaillent en même temps à la résolution d'un problème. Un procédé de tri parallèle peut être représenté par un réseau de tri comme dans l'illustration plus haut. Un réseau de tri est composé d'arêtes directionnelles appelées « fils » qui sont représentés par des flèches, et de nœuds appelés « comparateurs » qui sont représentés par des cercles.

Lors de chaque tour, les deux livres reliés à un comparateur sont comparés et, si nécessaire, inversés. Plusieurs paires de livres, reliées à des comparateurs adjacents, peuvent être comparées en même temps. On peut suivre le chemin d'un livre d'un échange à l'autre l'amenant à la position souhaitée en suivant le fil qui lui correspond de haut en bas.

Mots clés et sites web

Tri parallèle, réseau de tri



- https://fr.wikipedia.org/wiki/R%C3%A9seau_de_tri
- https://fr.wikipedia.org/wiki/Algorithme_glouton





14. Soundex

Donald aimerait encoder des mots d'après leur prononciation. Il procède de la façon suivante :

- Garde la première lettre.
- Supprime A, E, I, O, U, H, W et Y parmi toutes les lettres suivant la première.
- Remplace les lettres suivantes comme suit :
 - B, F, P ou V → 1
 - C, G, J, K, Q, S, X ou Z → 2
 - D ou T → 3
 - L → 4
 - M ou N → 5
 - R → 6
- Si deux lettres ou plus encodées par le même nombre sont adjacentes dans le mot d'origine, ne retiens que la première des deux lettres. Cela vaut également pour la première lettre du mot.
- Ne garde que les quatre premiers signes (y compris la première lettre) en complétant si nécessaire par des zéros.



Les mots suivants sont encodés comme suit :

Euler → E460
Gauss → G200
Heilbronn → H416
Kant → K530
Lloyd → L300
Lissajous → L222

Quel est le code pour le mot « Hilbert » ?

- A) H410
- B) B540
- C) H041
- D) H416



Solution

La première lettre étant un H, le premier signe du code est également H.
Tous les A, E, I, O, U, H et W sont ensuite enlevés, il ne reste donc que « lbrt » à traduire.
En remplaçant les lettres par le chiffre correspondant, on obtient H4163.
Il ne faut rien éliminer car il n'y a pas de lettres doubles.
En ne gardant que les quatre premiers signes, on obtient H416.

C'est de l'informatique !

Le procédé Soundex, plus exactement le Soundex américain, a été développé et patenté il y a 100 ans par Robert C. Russel et Margaret King Odell. Il a été utilisé pour trouver des mots, en particulier des noms propres, à consonance similaire dans la langue anglaise. Cela fonctionne parce que les groupes de lettres encodées par un même chiffre ont une phonétique similaire : en anglais, B, F, P et V sont des consonnes bilabiales, C, G, J, K, Q, S, X et Z des labio-dentales, D et T des dentales, L une alvéolaire, M et N des vélaires et R une laryngale.

Ce procédé étant très simple et donnant de relativement bons résultats également dans d'autres langues que l'anglais (en faisant éventuellement quelques modifications au code pour refléter la phonétique de la langue), il est souvent utilisé pour la recherche phonétique, c'est-à-dire la recherche de mots prononcés similairement. Il est inclus comme standard dans beaucoup de bases de données.



Les exemples cités plus haut viennent de Donald Knuth, un des grands informaticiens du 20^e siècle, qui travaille encore aujourd'hui à son livre « The Art of Computer Programming ». Le procédé décrit ici se trouve dans le troisième volume, « Sorting and Searching ».




Mots clés et sites web



Recherche phonétique, Soundex



- <https://www.functions-online.com/soundex.html>
- <https://fr.wikipedia.org/wiki/Soundex>
- <https://www-cs-faculty.stanford.edu/~knuth/taocp.html>
- <http://www.highprogrammer.com/alan/numbers/soundex.html>

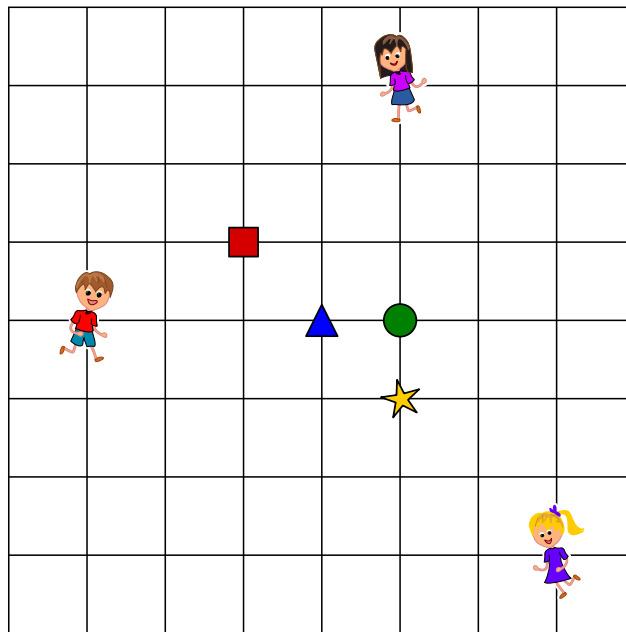


15. Trois amis





Alice , Bob  et Céline  habitent à la Chaux-de-Fonds. Ils ont marqué leurs domiciles sur le plan. Ils aimeraient trouver un lieu de rendez-vous pour lequel la somme de leurs distances de trajet est la plus petite possible. La distance de trajet est calculée en additionnant le nombre de segments entre deux intersections.

, ,  et  sont des lieux de rendez-vous possibles. Par exemple, la distance de trajet

la plus courte pour Alice  jusqu'au lieu de rendez-vous  est 4, car il y a quatre segments entre l'intersection où Alice se trouve et celle où le triangle se trouve.





Quel est le lieu de rendez-vous pour lequel la somme des distances de trajet des trois amis est la plus courte ?


- | A) | B) | C) | D) |
|---|---|---|--|
|  |  |  |  |




Solution

La bonne réponse est C) . La somme des distances de trajet des amis jusqu'au rond vert est $3 + 4 + 5 = 12$.

Mauvaise solution : . La somme des distances de trajet des amis jusqu'au carré rouge est $4 + 3 + 8 = 15$.

Mauvaise solution : . La somme des distances de trajet des amis jusqu'au triangle bleu est $4 + 3 + 6 = 13$.

Mauvaise solution : . La somme des distances de trajet des amis jusqu'à l'étoile jaune est $4 + 5 + 4 = 13$.

C'est de l'informatique !

Afin de déterminer lequel des quatre lieux de rendez-vous est le meilleur, il faut calculer la somme des distances de trajet des trois amis pour chacun d'entre eux. Le lieu de rendez-vous ayant la plus petite somme est le meilleur. C'est facile et ça ne prend pas beaucoup de temps – le calcul peut même se faire mentalement. Mais le même problème pour un grand nombre d'amis et un grand nombre de lieux de rendez-vous possibles devient un problème d'optimisation qu'on ne peut en général pas résoudre en un temps raisonnable.

Il existe en informatique des procédés permettant de trouver une solution proche de la solution optimale en un temps raisonnable pour ce type de problème. La solution ne diffère alors de la solution optimale que de moins de 1%, par exemple. S'il s'agit, au lieu d'amis se retrouvant, de livreurs de journaux allant chercher les journaux à distribuer à un endroit commun, 6 secondes de plus ou de moins ne font pas grande différence sur un trajet de 10 minutes (6 secondes représentent 1% de 10 minutes).

La recherche locale est un exemple de procédé pour trouver une solution presque optimale : afin de trouver un lieu de rendez-vous pour un grand nombre d'amis, on commence lors d'une recherche locale par n'importe quel lieu de rendez-vous qui peut être choisi aléatoirement. On compare ensuite la somme des distances de trajet pour ce lieu et les lieux de rendez-vous voisins et détermine le meilleur d'entre eux. On peut de cette manière se rapprocher de l'optimum.

Mais pourquoi les trois amis habitent-ils à la Chaux-de-Fonds ? La Chaux-de-Fonds et Le Locle font partie du patrimoine mondiale de l'humanité de l'UNESCO depuis 2009, pas uniquement en raison de l'histoire horlogère, mais également parce que les deux villes ont été reconstruites au 19^e siècle suite à des incendies selon un plan en damier, similaire à celui de la donnée de l'exercice.

Mots clés et sites web

Problème d'optimisation, recherche locale

- [https://fr.wikipedia.org/wiki/Recherche_locale_\(optimisation\)](https://fr.wikipedia.org/wiki/Recherche_locale_(optimisation))
- <https://whc.unesco.org/fr/list/1302/>



A. Auteurs des exercices

 Andrea Adamoli	 Bent Halden	 Wolfgang Pohl
 Jared Asuncion	 Urs Hauser	 Ilya Posov
 Daphne Blokhuis	 Juraj Hromkovič	 Nol Premasathian
 Lucia Budinská	 Takeharu Ishizuka	 J.P. Pretti
 Špela Cerar	 Svetlana Jakšić	 Doris Reck
 Kessarapan Charoensueksa	 Zhang Jinbao	 Chris Roffey
 Kris Coolsaet	 Dong Yoon Kim	 Kirsten Schlüter
 Valentina Dagienė	 Vaidotas Kinčius	 Andrea Maria Schmid
 Christian Datzko	 Jia-Ling Koh	 Jacqueline Staub
 Susanne Datzko	 Regula Lacher	 Allira Storey
 Dilek Doğan	 Anh Vinh Le	 Gabrielė Stupurienė
 Marissa Engels	 Dimitris Mavrovouniotis	 Peter Tomcsányi
 Hanspeter Erni	 Karolína Mayerová	 Troy Vasiga
 Gerald Futschek	 Samart Moodleah	 Rechilda Villame
 Ionuț Gorgos	 Tom Naughton	 Eslam Wageed
 Shuchi Grover	 Henry Ong	 Pieter Waker
 Yasemin Gülbahar	 Péter Piltmann	 Michael Weigend
 Martin Guggisberg	 Zsuzsa Pluhár	 Magdalena Zarach



B. Sponsoring : Concours 2018


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

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

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

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

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

 i-factory (Musée des transports, Lucerne)

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

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

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

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



<http://www.oxocard.ch/>
OXOcard
OXON



<http://www.diartis.ch/>
Diartis AG



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



AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

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



<http://www.hepl.ch/>
Haute école pédagogique du canton de Vaud



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



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



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



C. Offres ultérieures

010100110101011001001001
0100000100101110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétésuissepourl'infor
matique dans l'enseignement//societàsviz
zeraperl'informaticanell'insegnamento

Devenez vous aussi membre de la SSIE

<http://svia-ssie-ssii.ch/la-societe/devenir-membre/>

et soutenez le Castor Informatique par votre adhésion

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

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