

SOINDEX?



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA



HEILBRONN → H416

KANT → K530

Exercices et solutions 2018

Années scolaires 5/6



LISSAJOUS → L222



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

CASTORO → C236

LAOYD → L300

Éditeurs :

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

BIBER → B160

GAUSS → G200

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
00101101010101001101010011
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

CASTOR → C236





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. 31.



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>



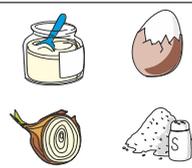
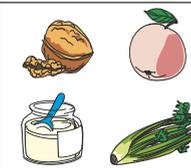
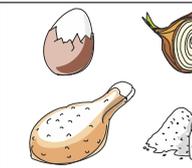
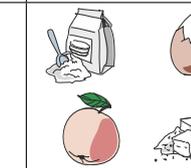
Table des matières

Ont collaboré au Castor Informatique 2018	i
Préambule	ii
1. Mets semblables	1
2. Colorier un motif	3
3. Serrure	7
4. Ensemble buissonnier	9
5. Les fleurs de Clara	11
6. Réseau de lignes	13
7. Planète Z	17
8. Glacier	19
9. Excursion avec vue	21
10. Les mensonges ne mènent pas loin	23
11. Chutes d'eau	27
12. L'étang des castors	29
A. Auteurs des exercices	31
B. Sponsoring : Concours 2018	32
C. Offres ultérieures	34



1. Mets semblables

Un cuisinier aimerait préparer deux mets. Ces deux mets ne doivent pas être semblables. Pour le cuisinier, deux mets sont semblables s'ils ont au moins deux ingrédients en commun.

pâtes	salade aux œufs	salade de noix	bouillon de poule	tourte
				

Quels mets sont semblables ?

- A) Le bouillon de poule et les pâtes
- B) Le bouillon de poule et la salade de noix
- C) Le bouillon de poule et la salade aux œufs
- D) La salade de noix et la tourte



Solution

La bonne réponse est C) Le bouillon de poule et la salade aux œufs.

Il y a de l'œuf, de l'oignon et du sel dans le bouillon de poule et dans la salade aux œufs.

Les autres combinaisons de mets ont au maximum un ingrédient en commun : le bouillon de poule et les pâtes contiennent les deux de l'oignon. Le bouillon de poule et la salade de noix n'ont pas d'ingrédient en commun, la salade de noix et la tourte non plus.

C'est de l'informatique !

Il y a beaucoup de situations dans lesquelles il faut comparer des choses et déterminer ce qui est semblable et ce qui est différent. Par exemple, les biologistes comparent le génome des bactéries, les chimistes les propriétés de substances, les astronomes comparent des galaxies, des étoiles et des planètes, et ainsi de suite.

Afin de comparer des choses, il faut définir quelles propriétés l'on compare. On peut ensuite déterminer à partir de quand deux choses sont semblables ou non. De cette manière, on peut par exemple dire qu'une chaise et une table sont semblables parce que les deux sont en bois. On peut tout aussi bien dire qu'une table n'est pas faite pour s'asseoir, et qu'une chaise n'est pas faite pour y écrire une lettre (même si ces deux choses sont bien sûr possibles). On peut aussi dire que deux chaises en bois ne sont semblables que si elles sont faites du même bois.

Pour cet exercice, il faut comparer cinq mets composés de quatre ingrédients chacun. Les biologistes, chimistes, astronomes et beaucoup d'autres scientifiques ne comparent pas un si petit nombre de choses, mais des milliers, millions ou milliards de choses qui peuvent posséder beaucoup de propriétés à considérer pour évaluer leur similarité. C'est ici que l'informatique entre en jeu en permettant la comparaison systématique d'un grand nombre de données d'après des critères de similarité prédéfinis.

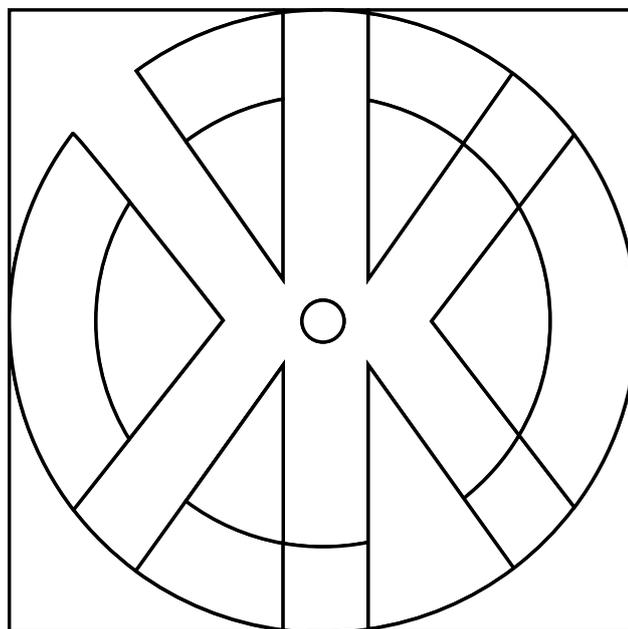
Mots clés et sites web

Objet, propriété, critère de similarité, Big Data

- [https://fr.wikipedia.org/wiki/Similarité_\(informatique\)](https://fr.wikipedia.org/wiki/Similarité_(informatique))
- https://fr.wikipedia.org/wiki/Big_data



2. Colorier un motif



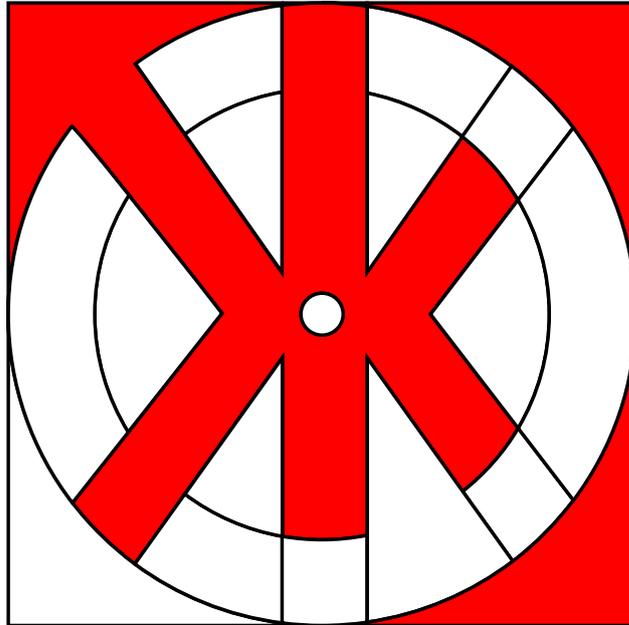
Les castors aimeraient bien colorier ce motif. Pour les aider, colorie les différentes surfaces du motif de sorte que chaque surface avoisinante présente une couleur différente. De plus, pour ce faire, choisis le moins de couleurs possibles.



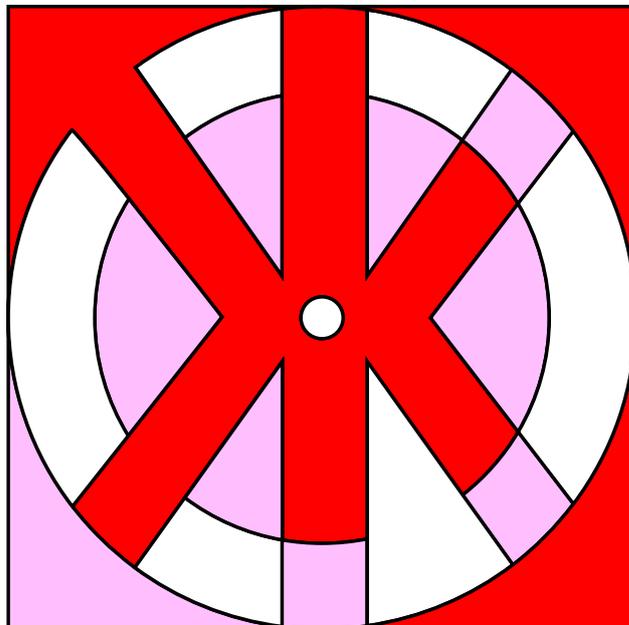
Solution

Il est possible de le faire à l'aide de trois couleurs différentes.

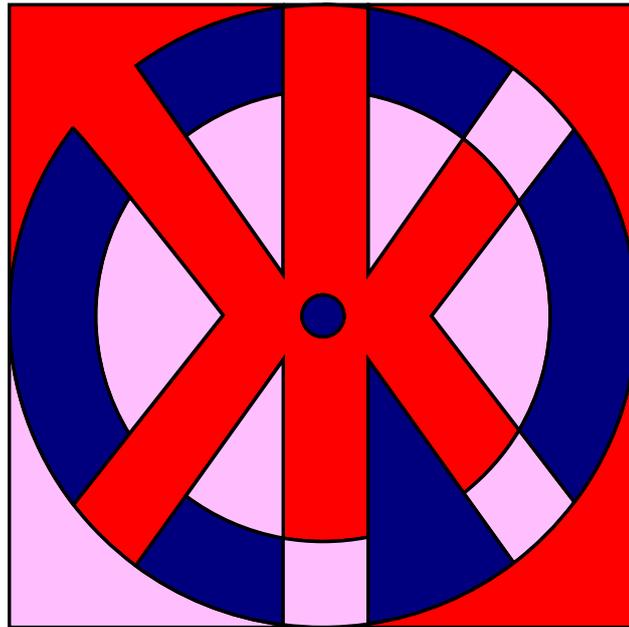
Il existe différentes solutions selon la première couleur que l'on choisit. Si l'on commence, par exemple, avec la couleur rouge dans l'angle supérieur gauche et que l'on colorie ensuite de la même couleur toutes les surfaces qui ne sont pas en contact direct avec elle, on obtient le résultat suivant :



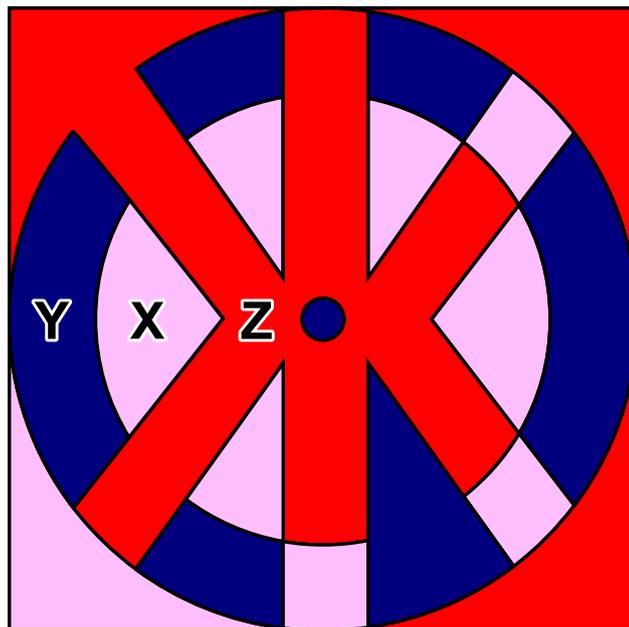
Si l'on continue avec une deuxième couleur, par exemple le rose, dans l'angle inférieur gauche et que l'on colorie chaque surface possible, on obtient le résultat suivant :



En principe, si l'on admet que le blanc est une couleur, on a ainsi déjà terminé, car avec les trois couleurs, rouge, rose et blanc, on a colorié le motif en respectant les deux conditions définies. Bien sûr, on est libre de colorier les surfaces blanches avec une troisième couleur (par exemple le bleu) :



Par contre, il ne serait pas possible de colorier le motif en question avec moins de trois couleurs. La surface X jouxte la surface Y, elles doivent donc être coloriées en différentes couleurs. Toutes deux délimitent la troisième surface Z et donc la couleur de celle-ci doit impérativement différer des deux autres.



C'est de l'informatique !

Quel est le nombre maximum de couleurs dont on a besoin pour colorier un nombre arbitraire de surfaces de sorte qu'aucune surface adjacente ne présente la même couleur ? La réponse correcte est : quatre couleurs suffisent tant que l'on n'autorise pas les « enclaves ». Une enclave est une zone partielle autonome qui appartient à une autre zone, mais qui n'y est pas reliée, comme par exemple la commune fribourgeoise d'Estavayer qui est, avec 11 autres communes, à l'intérieur du canton de Vaud, ou Büsingen sur le Rhin supérieur ou encore Campione d'Italia. À l'étranger, on pourrait aussi mentionner Baarle aux Pays-Bas et en Belgique.



Cette tâche présente le théorème des quatre couleurs qui est très difficile à prouver. Il y a 200 ans, on était arrivé à la conclusion que le nombre maximum était de cinq couleurs. Ce n'est qu'en 1976 que les mathématiciens Kenneth Appel et Wolfgang Haken ont prouvé qu'il suffisait de quatre couleurs. Ils ont utilisé des ordinateurs pour vérifier diverses exceptions et contre-exemples. Mais comme ce n'était plus possible de vérifier manuellement les calculs effectués par l'ordinateur, beaucoup de mathématiciens ont remis en cause l'utilisation de l'ordinateur pour ces types de preuves. Aujourd'hui encore, certains mathématiciens se demandent s'il est permis d'utiliser un ordinateur pour établir la véracité d'un énoncé mathématique ou pour prouver un théorème.

Dans la vie quotidienne, le théorème des quatre couleurs est très utile dans différents domaines, par exemple dans celui de l'aéronautique pour créer des plans de vol lorsque les avions sont assignés à des couloirs afin qu'ils aient toujours une distance suffisante, ou encore dans celui de la téléphonie mobile lorsque l'on assigne des plages de fréquences aux antennes de téléphonie mobile afin qu'elles n'interfèrent pas entre elles et que la réception ne se détériore pas malgré les nombreuses antennes.

Mots clés et sites web

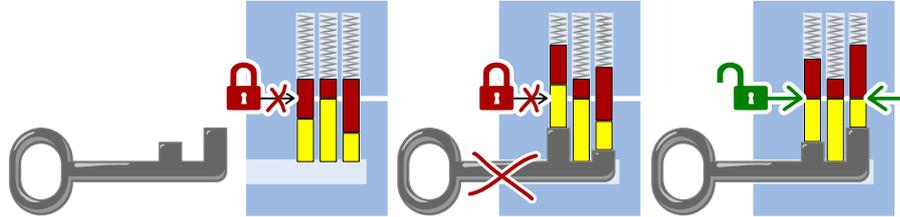
le théorème des quatre couleurs

- https://fr.wikipedia.org/wiki/Théorème_des_quatre_couleurs
- <http://www.mathepedia.de/Vier-Farben-Satz.html>
- https://en.wikipedia.org/wiki/Enclave_and_exclave
- <https://en.wikipedia.org/wiki/Baarle>

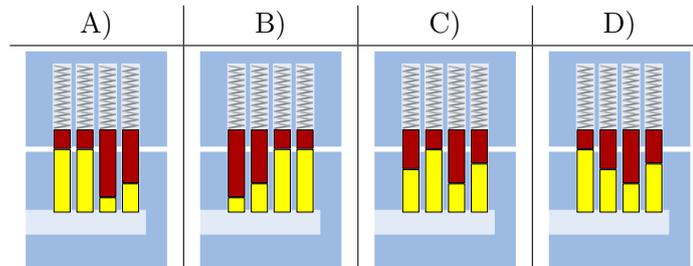


3. Serrure

Henry travaille chez un serrurier. Les serrures fonctionnent de la manière suivante :



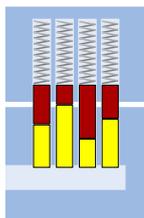
Quelle serrure peut être ouverte avec la clé suivante ?



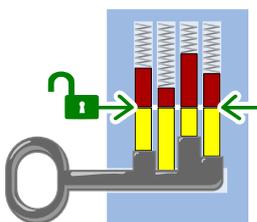


Solution

La bonne réponse est C) :



Lorsque la clé est introduite, les crans de la clé poussent toutes les goupilles de la serrure à la même hauteur de façon à ce que le cylindre puisse être tourné :



C'est de l'informatique !

Pour qu'une clé puisse ouvrir et fermer une serrure, il faut que chacun de ses membres corresponde aux éléments de la serrure. Pour cela, les longs crans de la clé doivent se trouver à la hauteur des courtes goupilles de la serrure, et les crans courts à la hauteur des longues goupilles. Les deux motifs doivent correspondre l'un à l'autre. Dans cet exercice, c'est le cas lorsqu'ils sont exactement opposés. La recherche d'un motif ou d'une forme spécifique est une tâche fondamentale de l'informatique, par exemple la recherche d'un mot dans un texte ou d'images similaires.

Mots clés et sites web

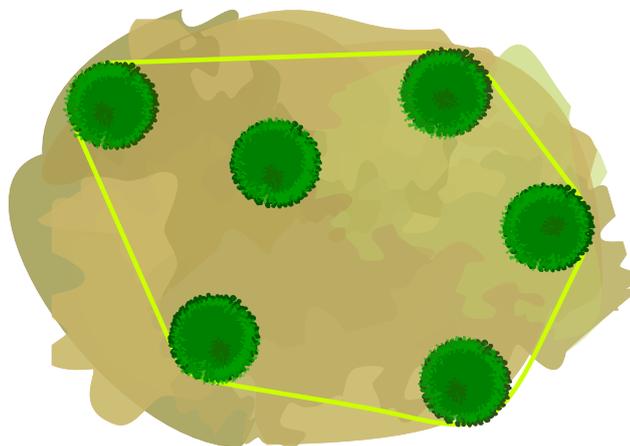
Reconnaissance de formes, serrure

- https://fr.wikipedia.org/wiki/Serrure_à_goupilles
- https://fr.wikipedia.org/wiki/Reconnaissance_de_formes

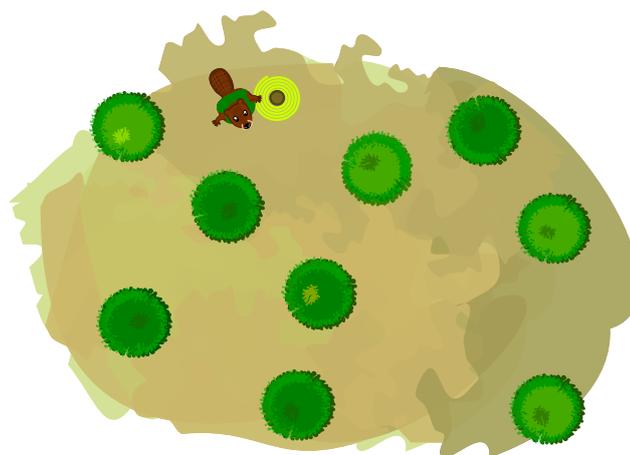


4. Ensemble buissonnier

Les castors entourent d'une corde le groupe de buissons qu'ils veulent abattre. Hier, ils voulaient abattre six buissons. La corde ne touchait que cinq des buissons. Voilà à quoi cela ressemblait vu du ciel :



Aujourd'hui, les castors veulent abattre ces neuf buissons :



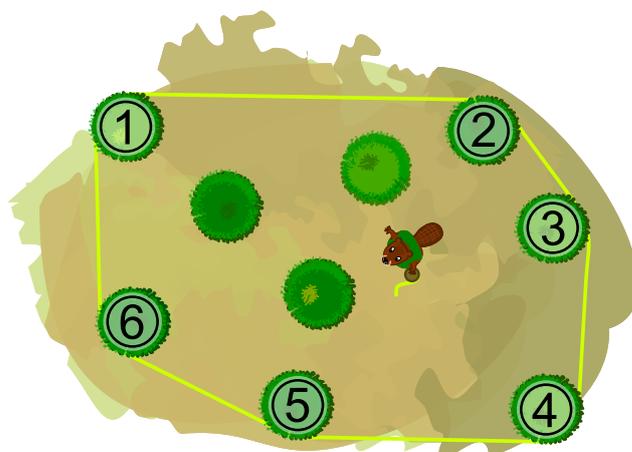
Combien de buissons la corde touche-t-elle cette fois-ci ?

- A) 3 buissons
- B) 4 buissons
- C) 5 buissons
- D) 6 buissons
- E) 9 buissons



Solution

La réponse D) est correcte. Les castors tendent la corde autour des buissons de la manière suivante :



La corde touche les six buissons numérotés.

C'est de l'informatique !

La corde entourant les buissons délimite la plus petite surface sur laquelle se trouvent tous les buissons qui doivent être abattus. La seule restriction est que les limites de la surface doivent être des lignes droites. Si les buissons sur l'image étaient des points, la corde aurait la forme d'un hexagone.

Le plus petit polygone contenant tous les points d'un ensemble donné est appelé l'*enveloppe convexe* de cet ensemble de points. Ici, *convexe* est utilisé pour quelque chose qui s'étend vers l'extérieur comme la lentille convexe d'une loupe. Une *enveloppe* est quelque chose qui entoure autre chose sans être plus grand que nécessaire, comme la peau entoure le corps. La corde des castors entoure donc les buissons comme une enveloppe convexe.

En informatique, on détermine souvent quelle est l'enveloppe convexe d'un ensemble de points :

- Reconnaissance de forme : l'image contient-elle un visage ?
- Reconnaissance de l'écriture manuelle : est-ce qu'un signe écrit est la lettre B ?
- Systèmes d'information géographique : quelle est la taille d'une plaine d'inondation ou d'un réseau de drainage ?
- Emballage : Quelle est la plus petite quantité de matériel nécessaire à l'emballage d'un objet ?

Il existe des méthodes informatiques qui déterminent l'enveloppe convexe d'un ensemble de points de manière efficace. Ces méthodes fonctionnent également pour un très grand nombre de points.

Mots clés et sites web

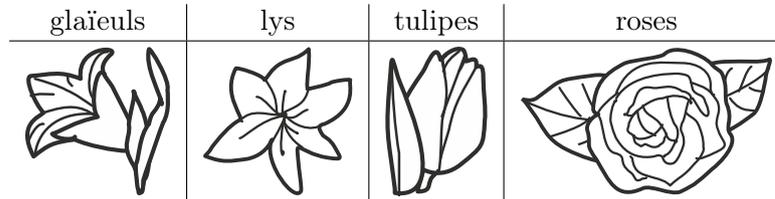
Graphe, enveloppe convexe

- https://fr.wikipedia.org/wiki/Enveloppe_convexe
- https://fr.wikipedia.org/wiki/Calcul_de_l'enveloppe_convexe
- <https://brilliant.org/wiki/convex-hull>



5. Les fleurs de Clara

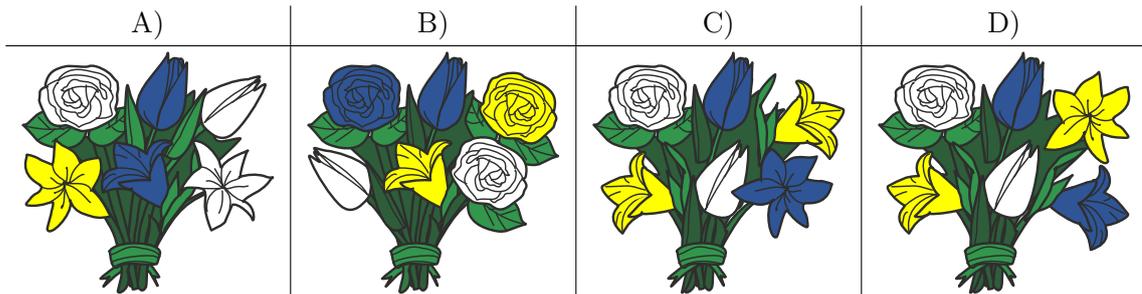
Clara va chez le fleuriste, car elle aime les bouquets de fleurs colorés. Elle y trouve les sortes de fleurs suivantes :



Chaque sorte de fleur est disponible en trois couleurs : blanc, **bleu** et **jaune**. Clara aimerait un bouquet de six fleurs qui remplit les conditions suivantes :

1. Il doit y avoir deux fleurs de chaque couleur (blanc, bleu, jaune),
2. Les fleurs de la même sorte ne doivent jamais être de la même couleur,
3. Il ne doit pas y avoir plus de deux fleurs de la même sorte.

Quel est le bouquet qui remplit les trois conditions ?





Solution

La bonne réponse est D). Le bouquet A) contient trois fleurs blanches (contrevient à la règle 1), le bouquet B) trois roses (contrevient à la règle 3), et le bouquet C) deux glaïeuls jaunes (contrevient à la règle 2).

C'est de l'informatique !

Les problèmes informatiques courants sont décrits par un ensemble de restrictions, la tâche étant de trouver une solution qui respecte toutes ou le plus possible de ces restrictions.

On peut considérer des tâches plus complexes lors desquelles des opérateurs logiques comme le connecteur ET (A ET B signifie que les deux conditions A et B doivent être remplies, comme les trois règles dans notre exercice) ou le connecteur OU (A OU B signifie que seulement une des deux conditions doit être remplie).

Mots clés et sites web

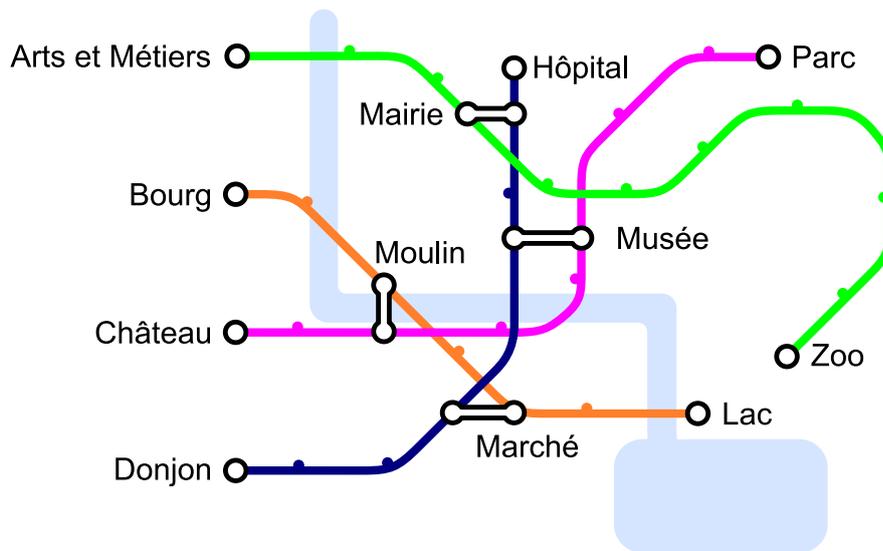
Conditions, opérateurs logiques

- <https://bookofbadarguments.com/>
- https://fr.wikipedia.org/wiki/Connecteur_logique
- <https://www.iep.utm.edu/prop-log/>



6. Réseau de lignes

Dans la ville des castors, il y a quatre lignes avec pour point de départ quatre stations différentes : les stations « Arts et Métiers », « Bourg », « Château » et « Donjon ». Chaque ligne comprend au moins une station de transit qui permet de changer de ligne : la station « Musée », la station « Marché », la station « Moulin » et la station « Mairie ».



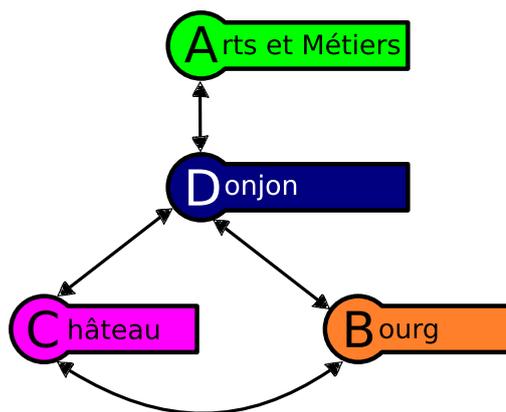
Aujourd'hui, Jean se rend au zoo. Il sait qu'il ne va changer de ligne qu'une seule fois. De quelle station de départ Jean est-il parti ?



Solution

La réponse correcte est « Donjon ». Quand on rebrousse chemin depuis la station « Zoo », on voit qu'il n'y a qu'une seule station de transit sur la ligne verte : « Mairie ». On en déduit que Jean est venu par la ligne bleue et que sa station de départ est « Donjon », étant donné qu'il n'a changé qu'une fois.

En informatique, il est possible de représenter le réseau de lignes à l'aide d'un graphe. Dans celui-ci, on représente les lignes par des nœuds et on les relie lorsqu'une station de transit permet de passer directement d'une ligne à l'autre. Par exemple, la ligne verte de « Arts et Métiers » à « Zoo », représentée par le A vert, est connectée à la ligne bleue de « Donjon » à « Hôpital », représentée par le D bleu, via la station de transit « Mairie » et donc le A est connecté au D dans ce schéma.



De cette ligne...	...on atteint ces lignes en un seul changement		
Arts et Métiers ↔ Zoo	Donjon ↔ Hôpital		
Bourg ↔ Lac	Château ↔ Place du parc	Donjon ↔ Hôpital	
Château ↔ Place du parc	Bourg ↔ Lac	Donjon ↔ Hôpital	
Donjon ↔ Hôpital	Bourg ↔ Lac	Château ↔ Place du parc	Arts et Métiers ↔ Zoo

Si on veut prendre la ligne « Arts et Métiers » ↔ « Zoo » pour arriver au zoo en ne changeant qu'une seule fois, on ne peut le faire qu'à partir de la station de départ « Donjon ». Ceci se lit dans le graphe en constatant que D est connecté uniquement à A, ou dans le tableau en voyant que A) n'apparaît qu'en regard de D), à la dernière ligne.

C'est de l'informatique !

Si tout cela te semble familier, c'est parce que beaucoup de réseaux de lignes de bus, de tramways ou de métros ressemblent à ce diagramme schématisé qui représente les lignes et les stations de ces transports publics. Il s'agit là d'une véritable invention : en 1931, Henry Beck a élaboré un diagramme schématisé pour le système de métro de Londres.

En informatique, un tel modèle abstrait est appelé un graphe. Un graphe se constitue par des nœuds (les stations) et des arêtes (le trajet entre deux stations). Dans notre tâche, il faut distinguer les nœuds qui présentent une ou deux arêtes (les stations de départ de fin ainsi que les stations intermédiaires) de ceux qui présentent un plus grand nombre d'arêtes (les stations de transit).



Dans la vie quotidienne, les graphes ont de nombreuses applications : les algorithmes élaborés d'un graphe peuvent par exemple résoudre des problèmes dans le domaine des réseaux sociaux, des guides routiers ou encore en matière de recherches sur les suggestions d'achats sur Internet. C'est la raison pour laquelle la maîtrise des graphes est une des compétences informatiques essentielles.

Mots clés et sites web

réseau de lignes, graphe

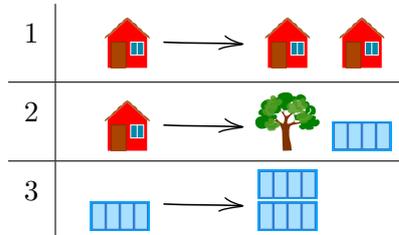
- https://fr.wikipedia.org/wiki/Plan_du_métro_de_Londres
- [https://fr.wikipedia.org/wiki/Graphe_\(mathématiques_discrètes\)](https://fr.wikipedia.org/wiki/Graphe_(mathématiques_discrètes))





7. Planète Z

Les habitants de la planète Z construisent toutes leurs villes de la même manière. Ils commencent chaque ville avec une maison, puis remplacent les bâtiments construits les uns après les autres en suivant les règles suivantes :

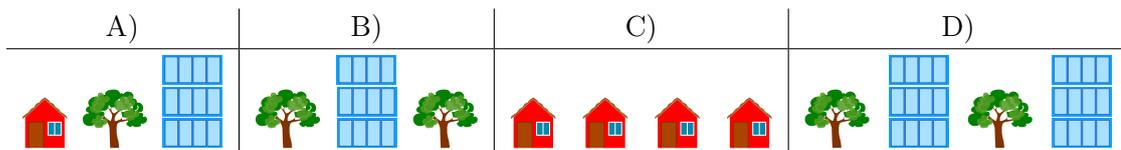


Par exemple, en appliquant d'abord la règle 1, puis la règle 2, puis deux fois la règle 3, on obtient la ville à droite de l'image ci-dessous :



L'ordre dans lequel sont arrangés les bâtiments et les arbres ne peut pas être modifié.

Laquelle des villes suivantes ne peut-elle pas se trouver sur la planète Z ?





Solution

La bonne réponse est B). Les arbres ne sont plantés dans la ville qu'en suivant la règle 2, et la règle 2 spécifie qu'il doit y avoir un immeuble à droite de chaque arbre. Dans la ville B), il n'y a pas d'immeuble à droite du deuxième arbre. Comme il n'existe pas de règle permettant d'enlever un immeuble, la ville B) ne peut pas être sur la planète Z.

La ville A) peut être construite en appliquant la suite de règles 1, 2, 3, et 3.

La ville C) peut être construite en appliquant trois fois la règle 1.

La ville D) peut être construite en appliquant d'abord la règle 1, puis la règle deux sur chacune des deux maisons. Finalement, on applique deux fois la règle 3 sur chaque immeuble.

C'est de l'informatique !

Les règles de cet exercice sont appelées règles de réécriture : un symbole ou un objet est remplacé par une série d'autres symboles ou objets. Si chaque règle ne remplace qu'un seul objet à la fois, le set de règles est appelé algébrique (ou non contextuel). Un symbole ou un objet est remplacé sans que le contexte (c'est-à-dire ce qui se trouve à sa gauche et à sa droite) ne joue de rôle.

En informatique, les règles de réécriture sont par exemple utilisées pour définir la syntaxe d'un langage de programmation. Les symboles ou objets sont des termes clés et les règles décrivent de quelle manière on peut les assembler en un programme (syntaxiquement) correct. Dans cet exercice, les symboles ou objets sont les maisons, les arbres et les immeubles. Les symboles ou objets et les règles de réécriture forment ensemble la grammaire décrivant un langage.

Lorsqu'un ordinateur traduit (compile) un programme en langage machine ou l'exécute directement (via un interpréteur pour les programmes écrits dans un langage de script), il commence par vérifier si le texte du programme suit bien les règles du langage de programmation. Il essaie donc, à l'aide d'un arbre syntaxique, de reconstruire les règles de réécriture qui transforment le symbole de départ (une maison dans cet exercice) en un texte de programme (les quatre réponses possibles dans cet exercice). Dans notre cas, c'est facilement réalisable, car les *mots* (des suites de symboles ou objets, dans cet exercice les villes) deviennent toujours plus grands et ne peuvent pas rétrécir.

Mots clés et sites web

Règle de réécriture, grammaire, langage algébrique

- [https://en.wikipedia.org/wiki/Production_\(computer_science\)](https://en.wikipedia.org/wiki/Production_(computer_science))
- https://fr.wikipedia.org/wiki/Langage_algébrique
- https://fr.wikipedia.org/wiki/Arbre_syntaxique
- https://fr.wikipedia.org/wiki/Grammaire_non_contextuelle
- https://fr.wikipedia.org/wiki/Règles_de_réécriture



8. Glacier

Il y a deux glaciers sur la place du village. Ils offrent les quatre mêmes sortes de glace :

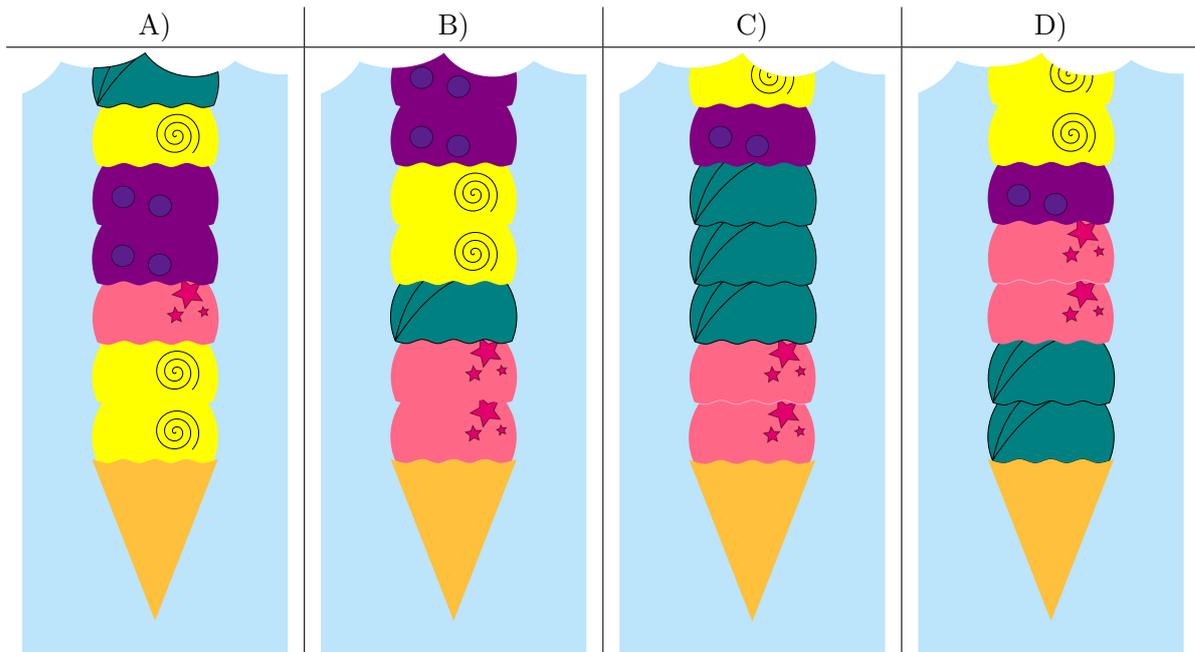


Le premier glacier suit les instructions suivantes pour préparer un cornet de glace :

1. Prends un cornet vide.
2. Choisis une sorte de glace au hasard et mets-en deux boules dans le cornet.
3. Ajoute une boule d'une des trois autres sortes de glace.
4. Si le nombre de boules souhaité est atteint, arrête. Sinon, recommence à l'étape 2.

Le deuxième glacier ne suit aucune règle.

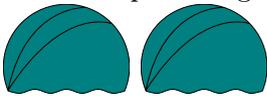
L'image suivante représente les premières boules de quelques cornets. Lequel provient à coup sûr du deuxième glacier ?





Solution

La bonne réponse est D). C'est le seul cornet qui ne peut pas avoir été préparé en suivant les instructions du premier glacier. Il commence en accord avec les règles avec deux boules de la même

sorte  suivies d'une boule d'une autre sorte ,

mais ensuite viennent deux boules de deux sortes différentes  , alors que les instructions spécifient que deux boules de la même sorte devraient être ajoutées.

Les réponses A), B) et C) ne sont pas correctes. Tous ces cornets pourraient avoir été préparés d'après les instructions du premier glacier.

C'est de l'informatique !

Une série d'instructions permet de créer des motifs dans des cornets de glace, des textes ou des images. Les informaticiennes et informaticiens développent des programmes informatiques permettant de reconnaître des motifs et des variations dans ces motifs. Parfois, des motifs sont générés par la ré-

pétition d'instructions. Par exemple, ce simple motif



est généré par la répétition de  suivi de . De tels motifs sont faciles à reconnaître. Le problème est plus compliqué dans cet exercice, car les instructions du premier glacier contiennent aussi des décisions laissées au hasard.

Il n'est de manière générale pas possible d'être sûr qu'une séquence a été générée par hasard ou en suivant une série d'instructions. Dans cet exercice, nous avons pu déterminer que l'un des cornets ne correspondait pas aux instructions et devait donc venir du deuxième glacier. On ne peut par contre jamais être sûr qu'une glace vient du premier glacier et pas du deuxième, car la composition correspondant aux instructions pourrait avoir été générée par hasard.

Mots clés et sites web

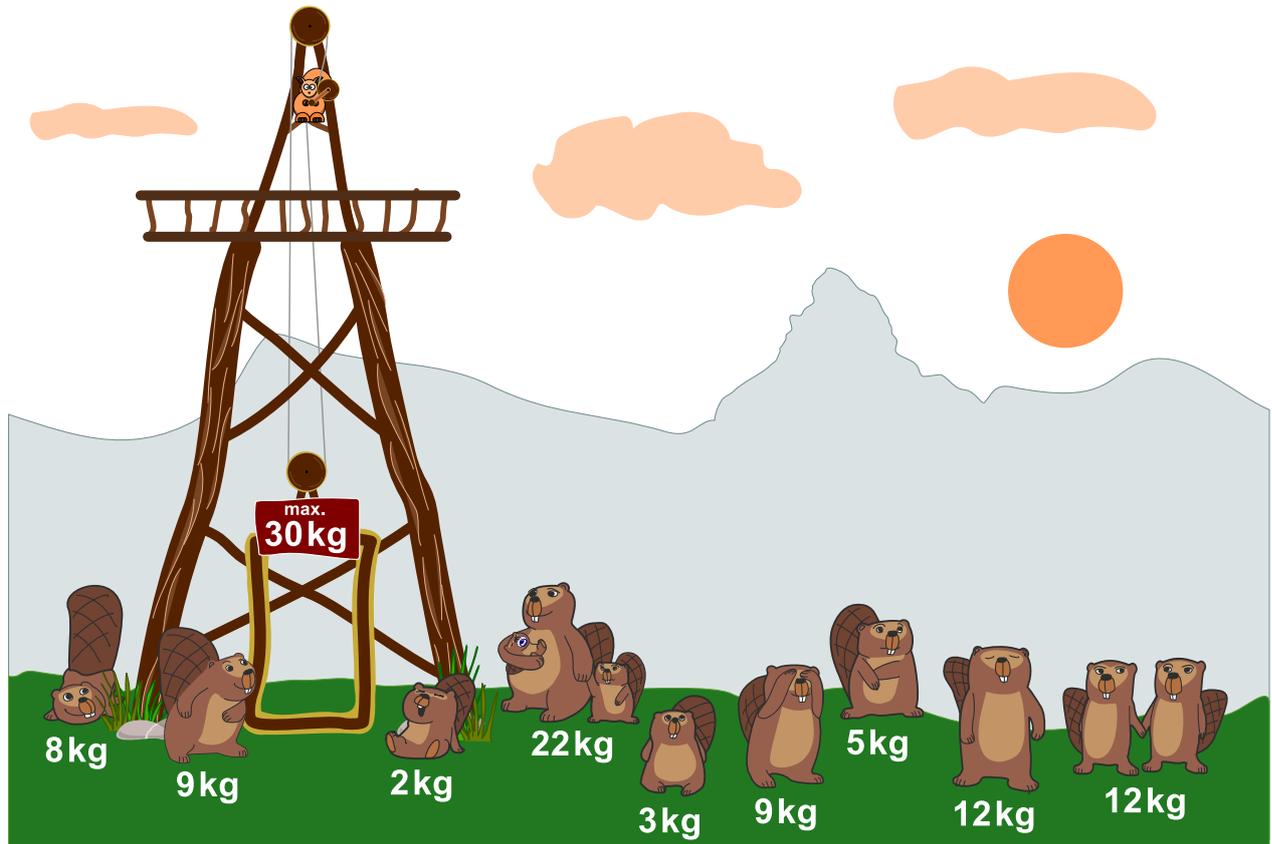
Reconnaissance de motifs

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



9. Excursion avec vue

Une famille de castors fait une excursion jusqu'à une tour panoramique. Ils sont en retard. L'ascenseur ne monte plus que deux fois avant la fermeture, et ne peut pas transporter plus de 30 kg d'un coup. Les jumeaux ne veulent monter qu'ensemble sur la tour. La maman castor porte le bébé dans ses bras et tient la main d'un petit castor. Et pourtant, on aimerait faire monter le plus grand nombre possible de castors au sommet de la tour.



Il faut se décider rapidement et seules les cinq options suivantes sont possibles. Qui doit rester en bas pour que le plus de castors possible puissent atteindre le sommet de la tour panoramique ?

- A) Tout le monde peut monter.
- B) La maman castor avec le bébé et le petit castor.
- C) Les jumeaux et le castor de 5 kg.
- D) Les jumeaux et la maman castor avec le bébé et le petit castor.
- E) La maman castor avec le bébé et le petit castor et le castor de 12 kg.

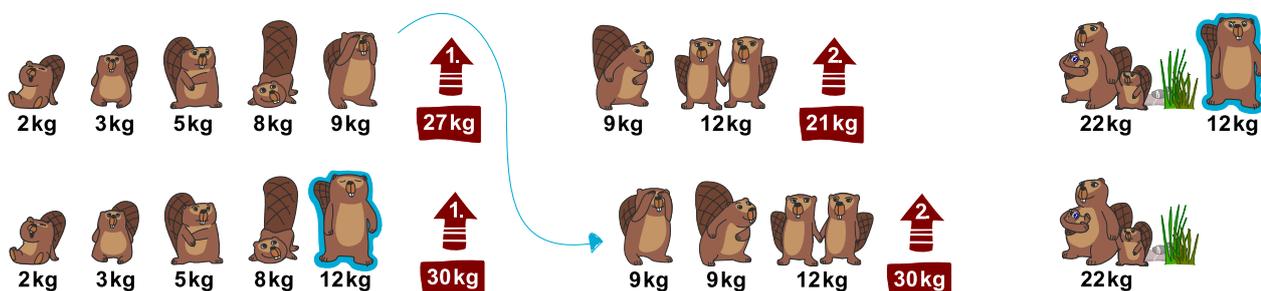


Solution

B) est l'une des bonnes solutions :



Une solution pourrait être de charger la cabine avec les castors les plus légers en premier : $2\text{ kg} + 3\text{ kg} + 5\text{ kg} + 8\text{ kg} + 9\text{ kg} = 27\text{ kg}$ lors de la première montée, et $9\text{ kg} + 12\text{ kg} = 21\text{ kg}$ lors de la seconde montée (cela fait 8 castors). Mais il y a encore de la place pour un castor supplémentaire dans l'ascenseur :



Cette stratégie permet d'utiliser les deux trajets en ascenseur de manière optimale : le castor de 9 kg est remplacé par celui de 12 kg lors du premier trajet. Le poids maximal de 30 kg est ainsi atteint. Le castor de 9 kg peut monter dans la cabine pour le second trajet, qui transporte également le poids maximal de 30 kg.

Les autres solutions proposées ne sont pas possibles : soit elles dépassent la charge maximale (le poids de la famille de castors entière dépasse 60 kg et même sans les jumeaux et le castor de 5 kg, elle atteint encore 65 kg), soit elles sont moins bonnes (si la maman avec les petits castors monte, ni les jumeaux, ni le castor de 9 kg, ni celui de 12 kg ne peuvent l'accompagner).

C'est de l'informatique !

Un des problèmes classiques en informatique est de trouver la combinaison optimale comme solution d'un problème. C'est souvent impossible de trouver une telle solution assez rapidement, ou dans un temps réaliste, parce qu'il y a trop de solutions possibles qui doivent être examinées. En informatique, on appelle cela un problème insoluble en pratique.

Ce problème est du même type que le *problème du sac à dos* lors duquel il faut ranger le plus d'objets possible dans un sac sans dépasser un poids maximal. Ce problème et les problèmes similaires sont appelés *NP-complets*. Ils ne peuvent être résolus que de manière approximative, ce qui veut dire que l'on peut trouver une bonne solution, mais pas nécessairement la solution optimale. On y parvient en réfléchissant à une stratégie judicieuse (« heuristique ») pour résoudre le problème.

Mots clés et sites web

Optimisation, problème du sac à dos

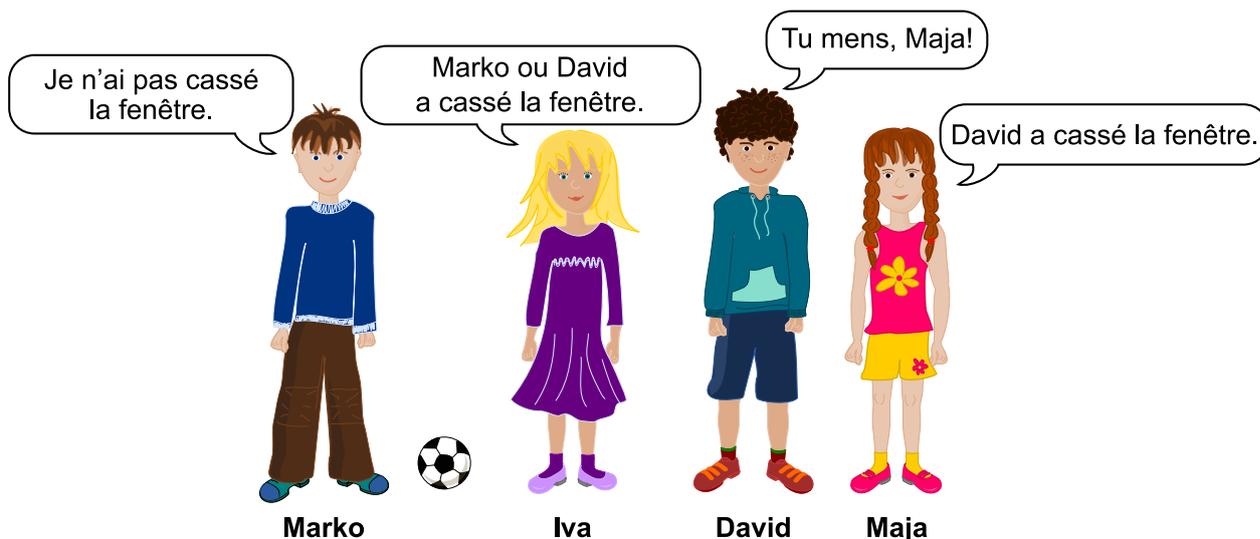
- https://fr.wikipedia.org/wiki/Optimisation_combinatoire
- https://fr.wikipedia.org/wiki/Problème_du_sac_à_dos



10. Les mensonges ne mènent pas loin

Par un jour de beau temps, Maja, David, Iva et Marko jouent au football près de la maison d'Anna. Tout à coup, une des fenêtres se casse et Anna cherche à savoir qui est responsable. Elle connaît les quatre enfants et sait que trois d'entre eux disent toujours la vérité; elle ne sait pas ce qu'il en est du quatrième.

Les quatre enfants disent :

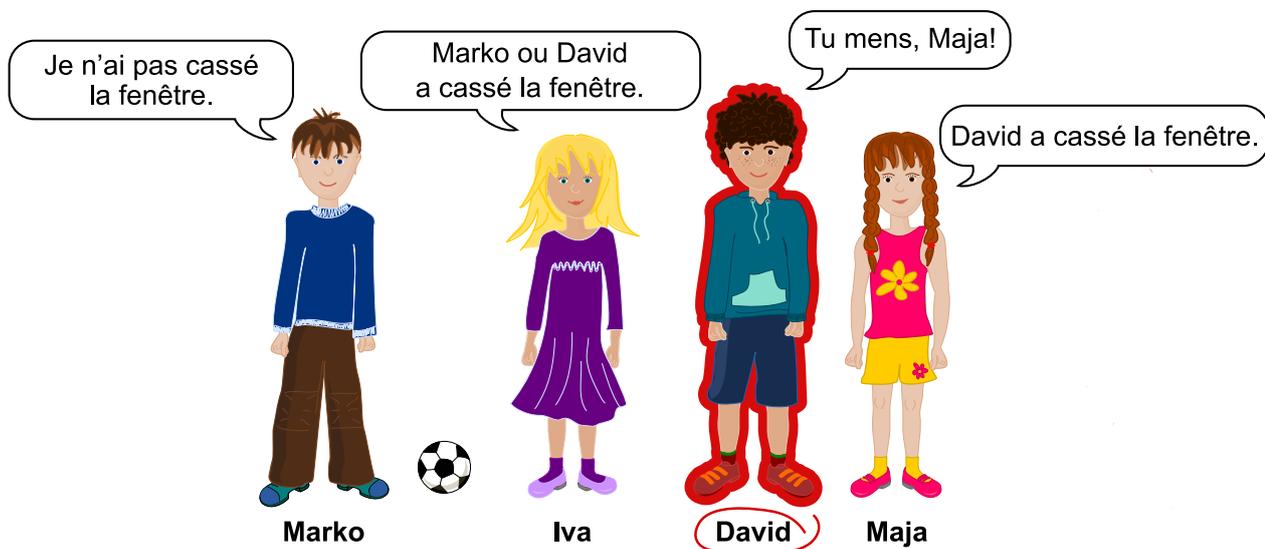


Clique sur l'enfant qui a cassé la fenêtre.



Solution

David a cassé la fenêtre.



Les déclarations de Maja et David ne peuvent pas être vraies toutes les deux ; l'un d'entre eux doit donc mentir. Si Maja disait la vérité, David mentirait et les déclarations d'Iva et de Marko seraient vraies. Par contre, si David disait la vérité, cela voudrait dire que Maja ment, mais aussi que soit Marko, soit Iva ment également, ce qui n'est pas possible étant donné que trois des enfants disent toujours la vérité.

C'est de l'informatique !

Tu dois raisonner de manière logique afin de résoudre ce problème. Le raisonnement se base sur la logique formulée en 1854 par George Boole (1815 – 1864), qui a décrit de manière formelle les bases d'énoncés logiques.

D'après lui, une déclaration (ou *assertion*) est soit *vraie* soit *fausse* (principe du tiers exclu). Plusieurs assertions peuvent être combinées à l'aide d'*opérateurs*. Des opérateurs logiques simples comme *ET* et *OU* lient deux assertions pour en former une nouvelle. Il existe aussi des opérateurs comme *NON* qui ne modifient qu'une seule assertion. La véracité de telles assertions combinées peut être déterminée à l'aide de *tables de vérité*.

L'un des opérateur qui représente la déduction « SI » → « ALORS » est l'implication. On parle alors de « tirer des conclusions logiques », c'est ce qui est nécessaire à la résolution de cet exercice.

Les ordinateurs fonctionnent également sur la base d'assertions booléennes et d'opérateurs logiques simples, car de tels ordinateurs peuvent être produits facilement en grandes quantités. Il existe quelques ordinateurs basés sur d'autres systèmes (par exemple les ordinateurs ternaires de la fin des années 50 en Union soviétique), mais soit ils sont restés au stade expérimental, soit ils n'ont jamais atteint une production de masse.

Mots clés et sites web

Logisches Schliessen

- https://fr.wikipedia.org/wiki/Principe_du_tiers_exclu
- https://fr.wikipedia.org/wiki/George_Boole
- [https://fr.wikipedia.org/wiki/Algèbre_de_Boole_\(logique\)](https://fr.wikipedia.org/wiki/Algèbre_de_Boole_(logique))



- [https://fr.wikipedia.org/wiki/Implication_\(logique\)](https://fr.wikipedia.org/wiki/Implication_(logique))
- https://fr.wikipedia.org/wiki/Ordinateur_ternaire
- https://it.wikipedia.org/wiki/Calcolatore_ternario

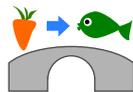




11. Chutes d'eau

 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.

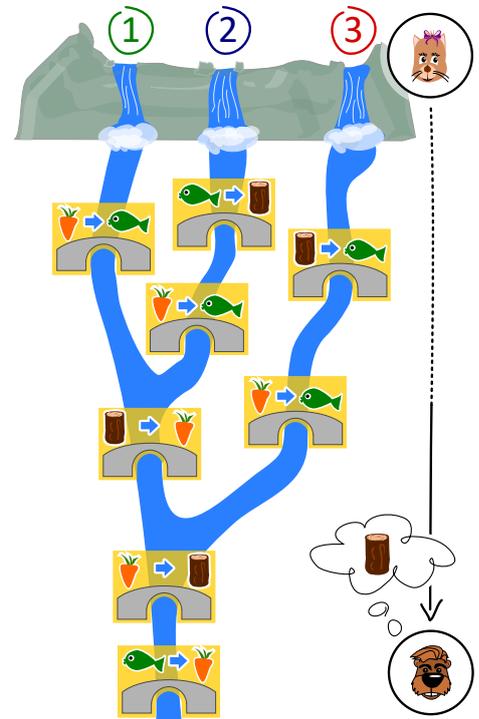
Katja peut lancer un poisson ou une carotte dans l'une des trois chutes d'eau. Les cours d'eau sont enjambés par plusieurs ponts sous lesquels vivent des trolls. Les trolls remplacent les objets passant sous les ponts par d'autres objets.



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.





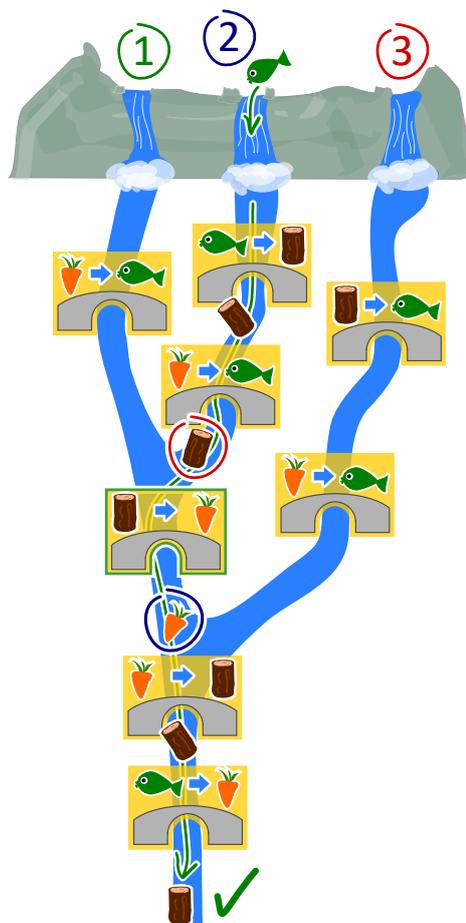
Solution

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

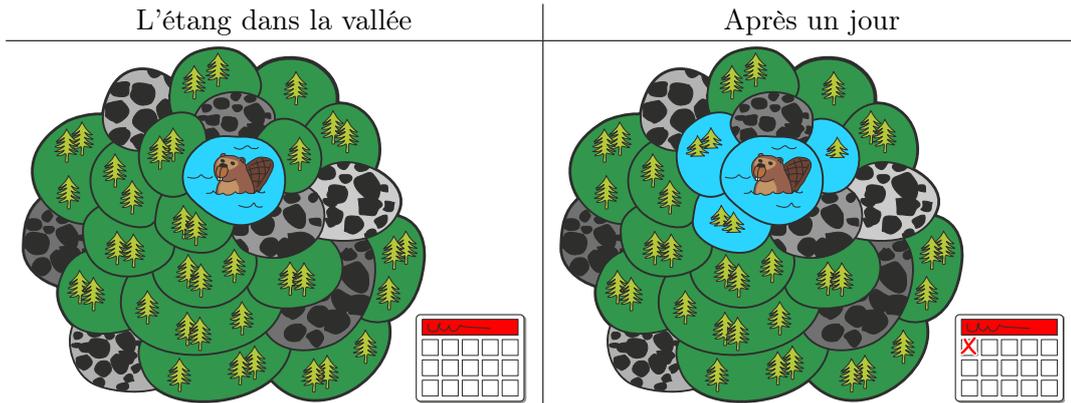
- [https://fr.wikipedia.org/wiki/Test_\(informatique\)](https://fr.wikipedia.org/wiki/Test_(informatique))
- https://fr.wikipedia.org/wiki/Boîte_blanche
- https://fr.wikipedia.org/wiki/Test_de_la_boîte_noire
- [https://fr.wikipedia.org/wiki/Paradigme_\(programmation\)](https://fr.wikipedia.org/wiki/Paradigme_(programmation))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



12. 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.



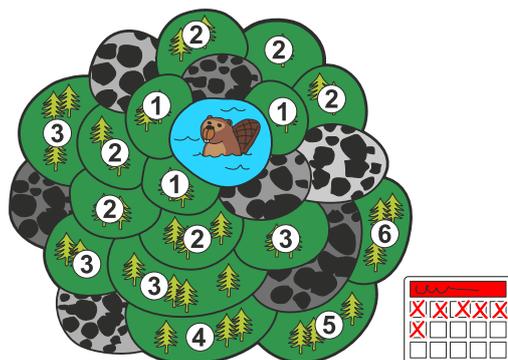
Après combien de jours en tout (y compris le premier jour représenté plus haut) les parcelles forestières sont-elles toutes inondées ?



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



A. Auteurs des exercices

 Jared Asuncion

 Javier Bilbao

 Laura Briviba

 Valentina Dagienė

 Darija Dasović Rakijašić

 Christian Datzko

 Susanne Datzko

 Hanspeter Erni

 Gerald Futschek

 Martin Guggisberg

 Urs Hauser

 Wei-fu Hou

 Juraj Hromkovič

 Takeharu Ishizuka

 Vaidotas Kinčius

 Regula Lacher

 Dan Lessner

 Dimitris Mavrovouniotis

 Anna Morpurgo

 Tom Naughton

 Sanja Pavlovic Šijanović

 Péter Piltmann

 Zsuzsa Pluhár

 Wolfgang Pohl

 J.P. Pretti

 Doris Reck

 Chris Roffey

 Andrea Maria Schmid

 Vipul Shah

 Mohamed El-Sherif

 Jacqueline Staub

 Allira Storey

 Faisal Al-Sudani

 Márta Szabó

 Willem van der Vegt

 Troy Vasiga

 Pieter Waker

 Michael Weigend

 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.



haute
école
pédagogique
vaud

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



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



Fachhochschule
Nordwestschweiz

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



Zürcher Hochschule der Künste
Game Design

<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.