



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

Exercices et solutions 2019 Années HarmoS 11/12

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

Éditeurs :

Gabriel Parriaux, Jean-Philippe Pellet, Elsa Pellet, Christian Datzko, Susanne Datzko, Juraj Hromkovič,
Regula Lacher

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





Ont collaboré au Castor Informatique 2019

Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Nora A. Escherle, Martin Guggisberg, Saskia Howald, Lucio Negrini, Gabriel Parriaux, Elsa Pellet, Jean-Philippe Pellet, Beat Trachsler.

Nous adressons nos remerciements à :

Juraj Hromkovič, Michelle Barnett, Michael Barot, Anna Laura John, Dennis Komm, Regula Lacher, Jacqueline Staub, Nicole Trachsler : ETHZ

Gabriel Thullen : Collège des Colombières

Valentina Dagienė : Bebras.org

Wolfgang Pohl, Hannes Endreß, Ulrich Kiesmüller, Kirsten Schlüter, Michael Weigend : Bundesweite Informatikwettbewerbe (BWINF), Allemagne

Chris Roffey : University of Oxford, Royaume-Uni

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

Gerald Futschek, Wilfried Baumann, Florentina Voboril : Oesterreichische Computer Gesellschaft, Austria

Zsuzsa Pluhár : ELTE Informatikai Kar, Hongrie

Eljakim Schrijvers, Justina Dauksaite, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Kyra Willekes, Saskia Zweerts : Cuttle.org, Pays-Bas

Christoph Frei : Chragokyberneticks (Logo Castor Informatique Suisse)

Andrea Leu, Maggie Winter, Brigitte Manz-Brunner : Senarclens Leu + Partner

La version allemande des exercices a également été utilisée en Allemagne et en Autriche.

L'adaptation française a été réalisée par Elsa Pellet et la version italienne par Veronica Ostini.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Le Castor Informatique 2019 a été réalisé par la Société Suisse de l'Informatique dans l'Enseignement SSIE et soutenu par la Fondation Hasler.

HASLERSTIFTUNG

Tous les liens ont été vérifiés le 1^{er} novembre 2019. Ce cahier d'exercice a été produit le 2 janvier 2020 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 en p. 48.



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 (années HarmoS 5 et 6) 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 2019 a été fait pour cinq tranches d'âge, basées sur les années scolaires :

- Années HarmoS 5 et 6 (Petit Castor)
- Années HarmoS 7 et 8
- Années HarmoS 9 et 10
- Années HarmoS 11 et 12
- Années HarmoS 13 à 15

Les élèves des années HarmoS 5 et 6 avaient 9 exercices à résoudre : 3 faciles, 3 moyens, 3 difficiles. Les élèves des années HarmoS 7 et 8 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 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 années HarmoS 7 et 8).

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

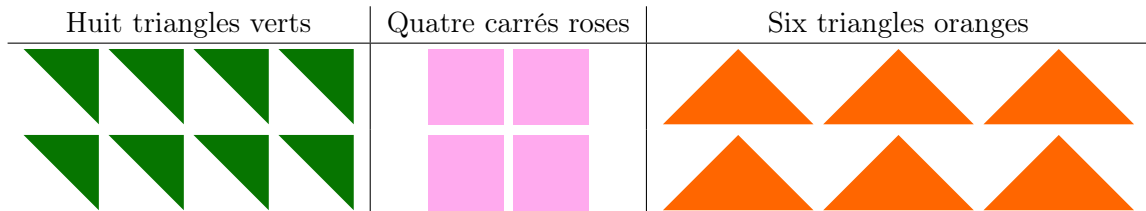
Ont collaboré au Castor Informatique 2019	i
Préambule	ii
Table des matières	iv
1. Rangoli	1
2. Drapeaux colorés	3
3. Caractères chinois colorés	5
4. Garniture de hamburger	9
5. Signaux de fumée	13
6. Tours particulières	15
7. Boules instables	17
8. Un sac de bonbons	21
9. Réseau de castors	25
10. Signaux lumineux	29
11. Quipu	33
12. Tempête de neige	35
13. Quel bonheur que les arbres !	37
14. Compression vidéo	41
15. Gare de triage	45
A. Auteurs des exercices	48
B. Sponsoring : Concours 2019	49
C. Offres ultérieures	51



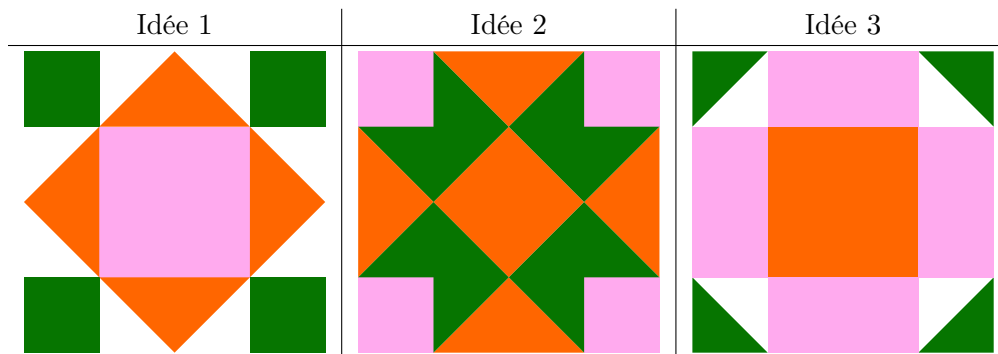
1. Rangoli

Le rangoli est une forme d'art venant d'Inde. Les rangolis sont des motifs faits sur le sol. Ces motifs sont le plus souvent symétriques.

Priya a des pierres de trois formes différentes pour son rangoli : huit triangles verts, quatre carrés roses et six triangles oranges. Les pierres de la même couleur ont la même taille :



Elle trouve les idées de rangoli suivantes sur un site Internet (les surfaces blanches restent vides) :



Laquelle des trois idées de rangoli Priya peut-elle faire avec ses pierres ?

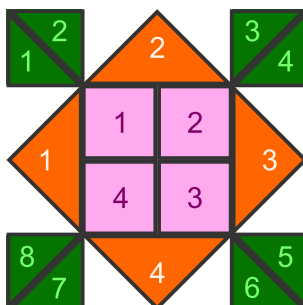
- A) Seulement l'idée 1.
- B) Seulement l'idée 2.
- C) Seulement l'idée 3.
- D) Les trois idées.



Solution

Priya ne peut faire que A) seulement l'idée 1 avec ses pierres.

L'illustration suivante compte les différentes pierres utilisées pour l'idée 1. Comme elle n'a besoin au maximum que du nombre de pierres qu'elle a à disposition, elle peut faire le motif de l'idée 1 :



Pour le motif de l'idée 2, elle aurait besoin de douze triangles verts, car chacune des quatre figures vertes de l'idée 2 nécessite trois triangles. Priya n'a que huit triangles verts et ne peut donc pas faire le motif de l'idée 2.

Pour le motif de l'idée 3, elle aurait besoin de huit triangles roses, car chacune des quatre figures roses de l'idée 3 nécessite deux carrés roses. Priya n'a que quatre carrés roses et ne peut donc pas faire le motif de l'idée 3.

Comme elle ne peut faire ni le motif de l'idée 2, ni celui de l'idée 3, la réponse D) ne peut pas être correcte.

C'est de l'informatique !

Le *rangoli* est une forme d'art venant d'Inde pour laquelle on utilise traditionnellement du riz et de la farine colorée, mais aussi du sable de couleur et des fleurs. Les rangolis ont surtout un but décoratif, mais sont aussi associés à des traditions locales et familiales ainsi qu'à des vœux de protection. Certaines traditions religieuses sont aussi associées aux rangolis.

Dans cet exercice, il fallait décomposer une forme complexe en formes plus petites que l'on pouvait comparer aux formes de base. On savait alors combien des formes de base étaient nécessaires. On appelle ce processus *décomposition*, il est souvent utilisé en informatique.

On appelle la comparaison des formes décomposées avec des formes de base *filtrage par motif*. Le filtrage par motif a une grande importance en informatique, il n'est pas utilisé que pour la recherche de motifs graphiques, mais aussi par exemple pour la recherche de mots dans un texte, de noms de fichiers dans un système de fichiers ou pour la comparaison de séquences génétiques lors de la recherche de criminels.

Mots clés et sites web

Décomposition, filtrage par motif

- <https://fr.wikipedia.org/wiki/Rangoli>
- https://fr.wikipedia.org/wiki/Filtrage_par_motif
- [https://en.wikipedia.org/wiki/Decomposition_\(computer_science\)](https://en.wikipedia.org/wiki/Decomposition_(computer_science))



2. Drapeaux colorés

Le constructeur de bateaux des castors fait d'excellents bateaux. Chaque castor veut en avoir un. Mais comment peuvent-ils faire la différence entre leurs bateaux s'ils sont tous pareils ?

Les castors décident de personnaliser chaque bateau avec un drapeau. Un drapeau des castors ressemble à cela :

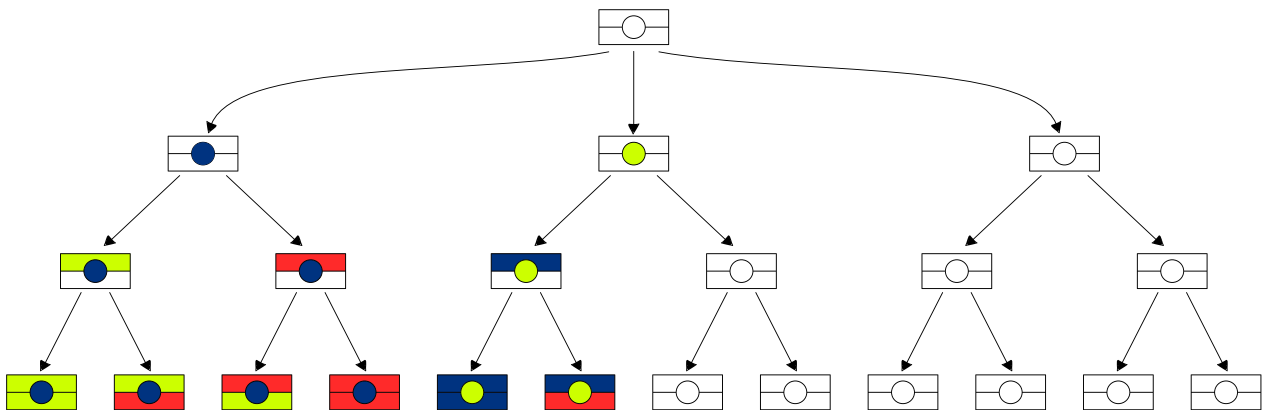


Ils se mettent d'accord sur trois couleurs pour les différentes surfaces du drapeau : rouge, vert clair et bleu foncé. Les deux bandes peuvent avoir la même couleur, mais le cercle au centre doit être d'une autre couleur que les deux bandes :



Pour ne pas perdre la vue d'ensemble, les castors dessinent un diagramme qui montre toutes les combinaisons de couleurs possibles pour les drapeaux, mais ils ne l'ont pas terminé.

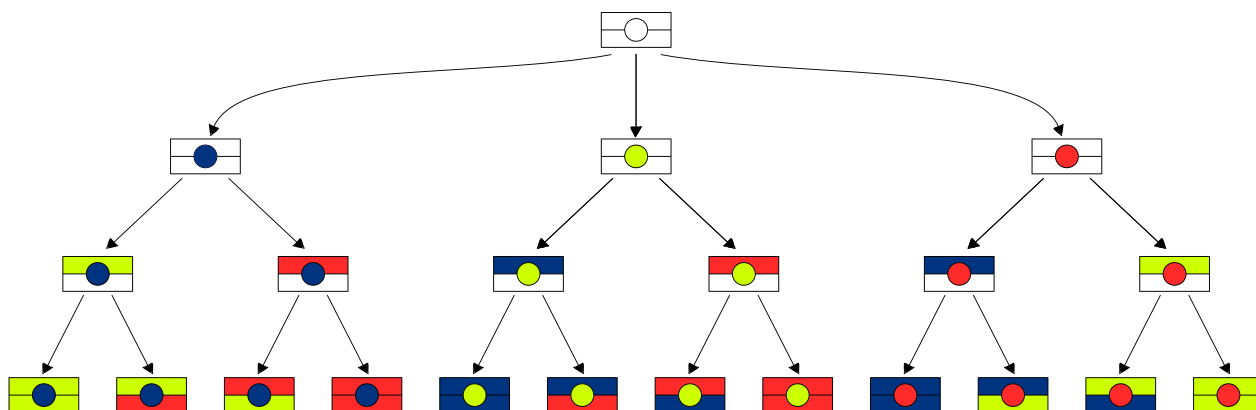
Complète le diagramme pour les castors. Il y a plusieurs bonnes solutions, cela suffit si tu en indiques une. Colorie les surfaces sur le diagramme.





Solution

Une bonne solution possible est :



De manière générale, les combinaisons du diagramme sont justes si...

- ... dans la deuxième rangée, le cercle à droite est rouge, ...
- ... dans la troisième rangée, en fonction de la couleur du cercle, la bande du haut a une couleur différente pour chaque couleur du cercle (l'ordre est égal), ...
- ... dans la quatrième rangée, en fonction de la couleur du cercle, la bande du bas a une couleur différente pour chaque couleur du cercle (l'ordre est égal).

C'est de l'informatique !

On doit parfois résoudre des problèmes compliqués. Pour cela, c'est utile de faire une liste de toutes les solutions possibles. En informatique, c'est spécialement important de pouvoir faire une liste de toutes les solutions possibles de manière efficace.

C'est souvent utile d'avoir une méthode systématique pour faire une liste afin de n'oublier aucune solution possible et de ne pas avoir de solution à double. Les structures de données comme l'*arbre* utilisé par les castors aident à trouver systématiquement toutes les solutions. Dans chaque rangée, toutes les valeurs possibles (ici les couleurs) d'une partie de l'objet (ici une surface du drapeau) sont dessinées côte à côte. On appelle le drapeau de la première rangée (pas coloré) la *racine*, et les drapeaux complètement colorés de la rangée du bas des *feuilles*. Un branchement est appelé *branche*. Comme chaque branches correspond à une valeur possible pour remplir la surface à colorier, on peut être sûr que toutes les solutions possibles sont sur les feuilles.

Mots clés et sites web

Arbre

- https://fr.wikipedia.org/wiki/Arbre_enracin%C3%A9
- <https://en.wikipedia.org/wiki/Enumeration>



3. Caractères chinois colorés

La structure des caractères chinois nous paraît étrange. Pour mieux comprendre comment certains caractères chinois sont assemblés, on peut s’imaginer le schéma suivant qui les sépare en cinq parties, haut ▲, bas ▤, gauche ▢, droite ▣ et centre ★ :



Ces parties peuvent être assemblées en quatre structures :

Structure	Structure gauche-centre-droite	Structure gauche-droite	Structure haut-centre-bas	Structure haut-bas
Exemple de caractère	川	儿	三	吕
Exemple d’analyse				

Quelle analyse montre la bonne structure pour les trois caractères chinois 劳, 二 et 八 d’après le schéma ?

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

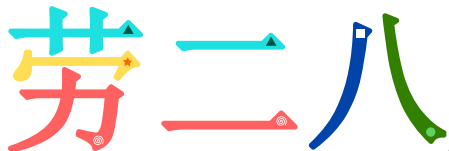


Solution

Le premier caractère, 劳, a une structure haut-centre-bas, le trait du haut est donc bleu clair , celui du centre jaune et celui du bas rose .

Le deuxième caractère, 二, a une structure haut-bas, le trait du haut est donc bleu clair et celui du bas rose .

Le troisième caractère, 八, a une structure gauche-droite, le trait de gauche est donc bleu foncé et celui de droite vert .



La bonne réponse est donc B)

Dans la réponse A), le deuxième caractère, 二, est bien analysé, mais les mauvaises couleurs sont associées aux caractères 劳 et 八: la couleur du haut est fausse pour 劳 et les deux couleurs sont inversées pour 八.

Dans la réponse C), tous les caractères sont mal analysés. Les couleurs du centre et du bas du premier caractère ne sont pas justes, celle du haut du deuxième caractère est fausse ainsi que les deux couleurs du troisième caractère.

Dans la réponse D), le caractère 八 est bien analysé, mais les couleurs du haut et du bas de 劳 sont fausses ainsi que les deux couleurs de 二.

C'est de l'informatique !

Les caractères chinois (*sinogrammes*) sont composés de caractères assemblés de manière complexe. Même les versions simplifiées ont plus de 200 éléments de base (*radicaux* ou *clés des sinogrammes*) différents, à partir desquels les caractères sont assemblés. Ceux-ci sont écrits l'un à côté de l'autre ou l'un en dessous de l'autre, formant des structures comme expliqué dans cet exercice. Ainsi, des milliers de caractères différents peuvent être assemblés. Si l'on doit apprendre à écrire avec ces caractères, il faut comprendre comment fonctionne l'assemblage. Pour cela, des couleurs sont souvent utilisées, comme dans cet exercice. L'alphabet latin utilisé chez nous fonctionne différemment: une lettre correspond à un son (avec des exceptions comme «eau», qui est prononcé [o] et non pas [eay]).

Qu'est-ce que cela a à voir avec l'informatique? Premièrement, de tels caractères doivent pouvoir être représentés à l'ordinateur. Il existe plusieurs approches pour cela, l'une d'entre elles utilise les radicaux décrits dans cet exercice. Deuxièmement, on doit avoir la possibilité de chercher des mots, par exemple dans un dictionnaire ou un lexique. Les radicaux les plus utilisés aujourd'hui viennent d'un dictionnaire qui a été élaboré entre 1710 et 1716 sous l'empereur Kangxi. Ce dictionnaire est ordonné d'après le nombre de traits de chaque radical.

Mots clés et sites web

Sinogramme

- https://fr.wikipedia.org/wiki/C1%C3%A9_d%27un_sinogramme
- https://fr.wikipedia.org/wiki/Sinogramme_simplifi%C3%A9
- https://fr.wikipedia.org/wiki/Codage_des_caract%C3%A8res_chinois
- https://fr.wikipedia.org/wiki/M%C3%A9thodes_de_saisie_et_d%27encodage_du_chinois
- https://fr.wikipedia.org/wiki/Dictionnaire_de_caract%C3%A8res_de_Kangxi
- https://fr.wikipedia.org/wiki/Alphabet_latin
- [https://fr.wikipedia.org/wiki/Eau_\(trigramme\)](https://fr.wikipedia.org/wiki/Eau_(trigramme))



Les caractères chinois sont :





- 川 : https://fr.wikipedia.org/wiki/Radical_47
- 儿 : https://fr.wikipedia.org/wiki/Radical_10
- 吕 : [https://en.wikipedia.org/wiki/L%C3%BC_\(surname\)](https://en.wikipedia.org/wiki/L%C3%BC_(surname))
- 二 : https://fr.wikipedia.org/wiki/Radical_7
- 三 : [https://fr.wikipedia.org/wiki/3_\(nombre\)](https://fr.wikipedia.org/wiki/3_(nombre))
- 八 : https://fr.wikipedia.org/wiki/Radical_12
- 劳 : <https://fr.wiktionary.org/wiki/%E5%8A%B3>



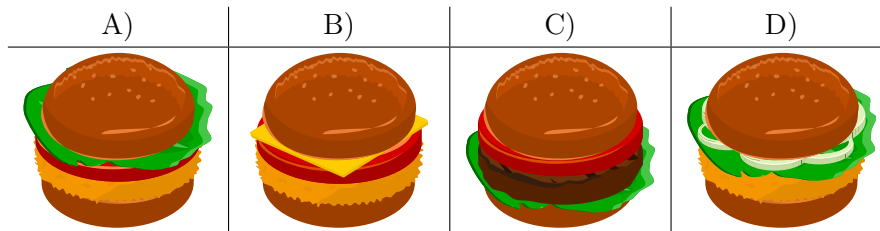


4. Garniture de hamburger

CastorBurger propose six ingrédients (A, B, C, D, E et F) pour ses hamburgers faits maison. Le tableau suivant liste les ingrédients de quatre exemples de hamburgers, pas forcément dans le même ordre que dans l'exemple de hamburger :

Hamburger				
Ingrédients	C, F	A, B, E	B, E, F	B, C, D

Quel hamburger a les ingrédients A, E et F ?





Solution

Pour déterminer à quel ingrédient correspond chaque lettre, il faut toujours comparer deux hamburgers :

Hamburgers comparés		Lettre en commun	Ingrédient en commun
		F	
		C	
		B	
		B (déjà identifié)	
		E	

Deux des ingrédients ne sont présents que dans un seul hamburger. Comme l'on a déjà identifié toutes les autres lettres, on peut identifier les ingrédients correspondants comme cela :

Hamburger particulier	Lettre particulière	Ingrédient particulier
	A	
	D	

Le hamburger recherché avec les ingrédients A, E et F doit donc contenir les ingrédients ,

et , et ce n'est le cas que du hamburger de la réponse A)





C'est de l'informatique !

L'*inférence* est la base de beaucoup de raisonnements, y compris en informatique. Pour résoudre cet exercice, on doit l'appliquer intensément : en comparant des hamburgers ayant des ingrédients en commun, on peut déduire des informations inconnues jusque-là (quel ingrédient correspond à quelle lettre).

Dans le cas de cet exercice, les ingrédients communs entre deux hamburgers correspondent à l'*intersection* des deux *ensembles* que sont les hamburgers. Elle ne contient que les ingrédients présents dans les deux ensembles. La première comparaison s'écrirait $\{C, F\} \cap \{B, E, F\} = \{F\}$. La contrepartie de l'intersection serait l'*union* $\{C, F\} \cup \{B, E, F\} = \{B, C, E, F\}$, elle contient tous les éléments qui sont présents dans au moins l'un des ensembles.

Pour déterminer quels ingrédients ne sont présents que dans un seul hamburger, on peut utiliser la *différence ensembliste*. Elle ne contient que les ingrédients du premier ensemble qui ne sont pas présents dans le deuxième ensemble. Par exemple, pour le premier hamburger particulier, on pourrait écrire : $\{A, B, E\} \setminus (\{C, F\} \cup \{B, E, F\} \cup \{B, C, D\}) = \{A, B, E\} \setminus \{B, C, D, E, F\} = \{A\}$.

La théorie des ensembles est peut-être connue du cours de mathématique. En informatique, elle est par exemple utilisée avec les bases de données. On peut aussi appliquer la théorie des ensembles telle quelle dans le domaine de la logique, appelé aussi algèbre de Boole, qui est utilisé souvent en informatique.

Mots clés et sites web

Inférence, théorie des ensembles, logique





- [https://fr.wikipedia.org/wiki/Inf%C3%A9rence_\(logique\)](https://fr.wikipedia.org/wiki/Inf%C3%A9rence_(logique))
- <https://fr.wikipedia.org/wiki/Ensemble>
- [https://fr.wikipedia.org/wiki/Alg%C3%A8bre_de_Boole_\(logique\)](https://fr.wikipedia.org/wiki/Alg%C3%A8bre_de_Boole_(logique))





5. Signaux de fumée

Un castor est toujours en haut de la montagne et observe la météo. Il transmet les prévisions météo aux castors dans la vallée. Pour cela, il utilise des signaux de fumée qui sont composés de cinq nuages de fumée. Un nuage de fumée peut être soit petit, soit grand. Les castors se sont mis d'accord sur les signaux de fumée suivants :

			
Ce sera orageux.	Ce sera pluvieux.	Ce sera nuageux.	Ce sera ensoleillé.

Un jour où il y a beaucoup de vent, les castors dans la vallée n'arrivent pas bien à reconnaître les nuages de fumée. Ils interprètent le signal de fumée comme cela :



Comme ce n'est aucun des signaux de fumée convenus, ils supposent qu'ils ont mal interprété l'un des nuages de fumée : l'un des petits nuages de fumée devrait en fait être grand ou l'un des grands nuages de fumée devrait en fait être petit.

Que voudrait dire le signal de fumée si exactement un nuage de fumée avait été mal interprété ?

- A) Ce sera orageux.
- B) Ce sera pluvieux.
- C) Ce sera nuageux.
- D) Ce sera ensoleillé.



Solution

Cinq signaux de fumée différents sont possibles si exactement un nuage de fumée a été mal interprété. Si l'on interprète le premier, deuxième, quatrième ou cinquième nuage de fumée différemment, on n'obtient aucun des quatre signaux de fumée convenus. Par contre, si l'on interprète le troisième nuage de fumée comme un petit nuage, cela donne la bonne réponse C) «Ce sera nuageux».





On peut aussi comparer le signal de fumée interprété aux quatre signaux de fumée convenus et regarder combien de nuages de fumée sont différents. Pour le signal «Ce sera orageux», deux nuages de fumée sont différents (le plus haut et le plus bas), pour le signal «Ce sera pluvieux», trois nuages de fumée sont différents (les deux plus hauts et le plus bas), pour le signal «Ce sera nuageux», un nuage de fumée est différent (celui du milieu, ce qui en fait la bonne réponse comme décrit ci-dessus) et pour le signal «Ce sera ensoleillé», quatre nuages de fumée sont différents (tous sauf le plus haut).

C'est de l'informatique !

Lorsque l'on doit transmettre un message, on aimerait que ce message arrive correctement à son destinataire. Les messages de cet exercice sont transmis à l'aide de petits et de grands nuages de fumée. Dans le cas général, on parle de *symboles*. C'est donc raisonnable de choisir une suite de symboles qui permette de comprendre le message même s'il est endommagé en cours de route. On peut faire cela en transmettant plus d'informations qu'il n'est strictement nécessaire. On appelle ces informations supplémentaires *redondantes*.

Lorsque l'on peut reconstruire un message avec au maximum n erreurs, on parle de code correcteur avec une capacité de correction n . La représentation de messages par des suites de symboles de manière à ce que l'on puisse les reconstruire même lorsque cette représentation a été endommagée lors de la transmission est une tâche typique pour les informaticiens. Ils nous permettent ainsi par exemple de lire de la musique à partir de CD ou des vidéos à partir de DVD même lorsque quelques erreurs ont eu lieu lors de la transmission.

Dans cet exercice, deux nuages de fumée auraient suffi pour transmettre les quatre messages différents :

			
Ce sera orageux.	Ce sera pluvieux.	Ce sera nuageux.	Ce sera ensoleillé.

Les castors utilisent cependant cinq nuages de fumée. Cela leur permet de comprendre le message même dans les cas où deux voire parfois trois des nuages de fumée sont «illisibles». Les castors ont de plus choisi les messages de manière à ce qu'il y ait au moins trois positions différentes entre chaque paire de messages.

Mots clés et sites web

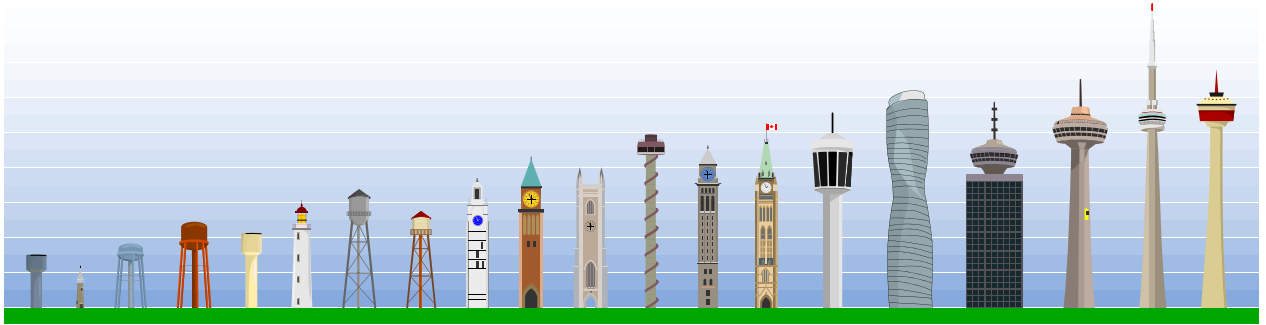
Code correcteur

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



6. Tours particulières

Une tour est considérée comme particulière lorsque toutes les tours à sa gauche sont plus petites qu'elle et toutes les tours à sa droite sont plus grandes qu'elle.



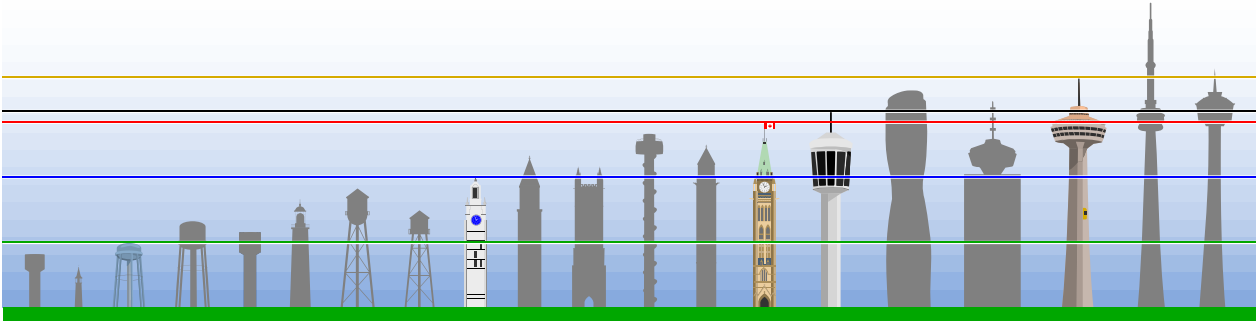
Combien de tours sont particulières sur le dessin ?

- A) 4
- B) 5
- C) 6
- D) 7



Solution

Comme on peut le voir aux lignes colorées, les cinq tours suivantes sont particulières, la réponse B) est donc correcte :



C'est de l'informatique !

Dans cet exercice, des tours sont comparées sur la base de leur hauteur. De telles comparaisons ont lieu lors de processus de *tri et recherche*, un domaine de l'informatique qui a été étudié en détail.

Il existe beaucoup d'algorithmes de tri différents qui sont adaptés à différentes applications. L'algorithme de *tri rapide* en est un exemple connu. L'identification de valeurs n'ayant que des valeurs plus petites à leur gauche et plus grandes à leur droite est un élément crucial de l'algorithme de tri rapide. Un tel élément partitionne le tableau à trier en deux sous-tableaux et divise ainsi le problème de tri de départ en deux problèmes plus petits. Cet élément entre les deux sous-tableaux est appelé *pivot*. Contrairement à cet exercice, dans l'algorithme de tri, les éléments à gauche et à droite du pivot ne sont pas plus petits et plus grand que le pivot dès le départ, mais doivent le devenir en procédant à des permutations d'éléments. Ce processus est répété pour chaque sous-tableau jusqu'à ce que chaque sous-tableau ne contienne plus qu'un seul élément... qui est déjà trié. Ce processus *récuratif* consistant à diviser un problème en problèmes plus petits avant de les résoudre est appelé *diviser pour régner*. C'est une manière très répandue de résoudre des problèmes difficiles.

L'algorithme de tri rapide est plus rapide que beaucoup d'autres algorithmes de tri, d'où son nom. Cela vient du fait que, dans le cas habituel, le choix du pivot permet de diviser la taille du tableau par deux. Un tableau de 1000 éléments nécessite donc environ 10 niveaux de division (mathématiquement, ce sont $\log_2(1000)$ niveaux de division). Étant donné que chaque élément doit être comparé au pivot, cela nécessite en tout 10 000 comparaisons. L'ordre de grandeur du nombre de comparaisons nécessaires en utilisant d'autres algorithmes répandus est plutôt de 1 000 000 comparaisons !

Mots clés et sites web

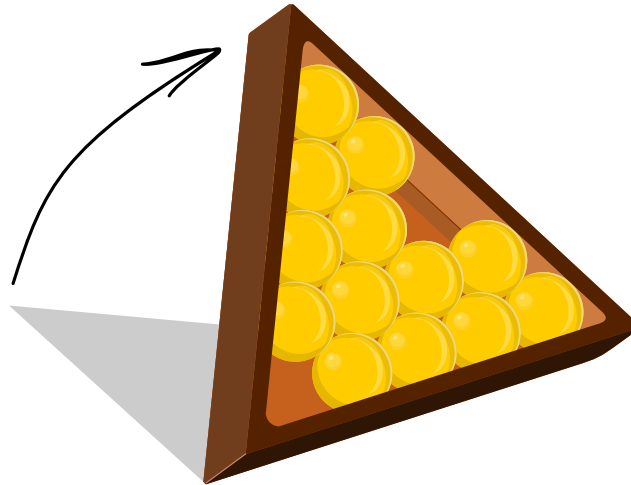
Tri rapide, pivot, diviser pour régner (Divide & Conquer)

- https://fr.wikipedia.org/wiki/Tri_rapide
- [https://fr.wikipedia.org/wiki/Diviser_pour_r%C3%A9gner_\(informatique\)](https://fr.wikipedia.org/wiki/Diviser_pour_r%C3%A9gner_(informatique))
- <https://www.youtube.com/watch?v=ywWBy6J5gz8>



7. Boules instables

Une boîte triangulaire peut contenir quinze boules de la même taille. Deux boules sont retirées de la boîte comme dans le dessin ci-dessous. La boîte est ensuite inclinée sur le côté.

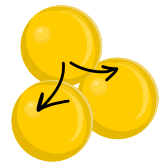


Lorsque l'on incline la boîte, certaines boules peuvent devenir « instables ». Un boule est instable lorsque...

- ... la boule à gauche ou à droite en dessous d'elle a été retirée, ...
- ... ou la boule à gauche ou à droite en dessous d'elle est instable.

Les boules de la rangée du bas sont stables.

Combien des treize boules sont instables ?

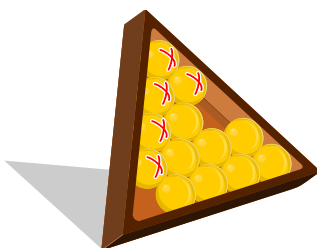


- | | | |
|-----------------|-------------|----------------------|
| A) Aucune boule | F) 5 boules | K) 10 boules |
| B) 1 boule | G) 6 boules | L) 11 boules |
| C) 2 boules | H) 7 boules | M) 12 boules |
| D) 3 boules | I) 8 boules | N) Toutes les boules |
| E) 4 boules | J) 9 boules | |



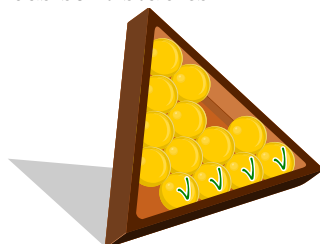
Solution

Cinq boules sont instables. Elles sont marquées d'une croix dans le dessin suivant :

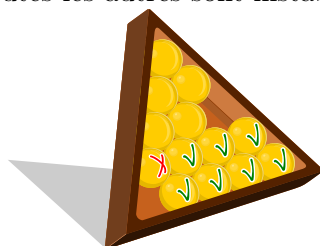


Le plus simple est d'y réfléchir en allant du bas vers le haut :

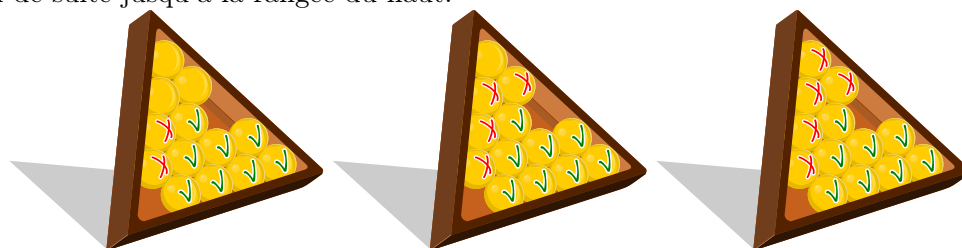
- Toutes les boules de la rangée du bas sont stables.



- Toutes les boules de la deuxième rangée depuis le bas qui ont deux boules stables en dessous d'elles sont également stables. Toutes les autres sont instables.



- Et ainsi de suite jusqu'à la rangée du haut.



C'est de l'informatique !

Il existe deux conditions pour qu'une boule soit classifiée comme instable. La première condition peut être testée directement. Afin de pouvoir tester la deuxième condition, on doit d'abord savoir si l'une des boules de la rangée en dessous est instable. C'est facile pour la rangée du bas, car comme il n'y a aucune boule en dessous, toutes les boules y sont stables. Comme expliqué dans la solution, on peut ensuite tester la rangée en dessus et déterminer quelles boules sont instables. De cette manière, on peut tester systématiquement chaque rangée du bas vers le haut et déterminer si chaque boule est instable.

Le principe d'après lequel une condition dépend du résultat d'une condition similaire s'appelle *récur-sivité*. Les conditions récursives sont construites de manière à ce que le résultat soit évident (*cas de base*, dans notre cas, toutes les boules de la dernière rangée sont stables) soit dépendant du résultat d'autres conditions récursives (*cas général*, dans notre cas, il s'agit de toutes les boules qui ne se



trouvent pas dans la dernière rangée et pour lesquelles il faut donc d'abord tester les boules du dessous).

Le principe de la récursivité est souvent utilisé en informatique. Il permet de résoudre de manière simple et élégante beaucoup de problèmes complexes. C'est également possible de transformer une solution récursive en une solution pas à pas (*itérative*). Un exemple classique pour lequel il existe une solution récursive simple est le jeu des tours de Hanoï.

Mots clés et sites web

Récursivité, tours de Hanoï

— <https://fr.wikipedia.org/wiki/R%C3%A9cursivit%C3%A9>

— https://fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF

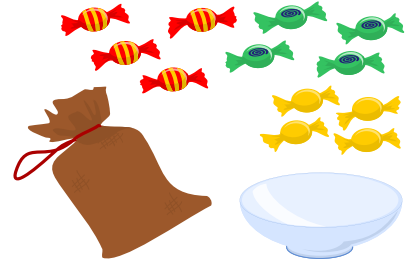




8. Un sac de bonbons

Petra a quatre bonbons rouges, quatre bonbons verts et quatre bonbons jaunes dans un sac opaque. Elle a aussi une coupe vide. Petra et Moritz jouent à un jeu. Pendant trois tours, Moritz peut tirer un bonbon du sac. Les règles suivantes valent pour chaque bonbon :

- Tant que le bonbon tiré est vert, il le met dans la coupe et peut tirer un autre bonbon pendant le même tour.
- Si le bonbon tiré est rouge, Moritz le met dans la coupe et termine le tour.
- Si le bonbon tiré est jaune, Moritz le mange directement sans le mettre dans la coupe et termine le tour.



Combien de bonbons au maximum Moritz peut-il avoir mis dans la coupe à la fin du jeu ?

- | | | |
|------|------|-------|
| A) 0 | F) 5 | K) 10 |
| B) 1 | G) 6 | L) 11 |
| C) 2 | H) 7 | M) 12 |
| D) 3 | I) 8 | |
| E) 4 | J) 9 | |



Solution

La bonne réponse est H) 7.

Dans le meilleur des cas, les quatre bonbons verts sont tirés. Cela signifie d'une part que les quatre bonbons verts sont dans la coupe et d'autre part que Moritz a pu tirer quatre fois un bonbon de plus au cours des trois tours, donc sept bonbons en tout.

Pour les trois bonbons restants, Moritz tire dans le meilleur des cas un bonbon rouge à chaque fois, qui sont ensuite aussi dans la coupe. Cela fait en tout quatre bonbons verts et trois bonbons rouges, donc sept bonbons dans la coupe en tout.

Il ne peut pas y avoir plus de sept bonbons dans la coupe. À chaque tirage, un bonbon au maximum peut être mis dans la coupe, et comme il n'y a que quatre bonbons avec lesquels on peut tirer un bonbon supplémentaire, cela fait au maximum sept bonbons.

L'ordre dans lequel les bonbons sont tirés dans le meilleur des cas est relativement égal tant que le dernier bonbon tiré est un bonbon rouge, car on peut dans ce cas toujours en tirer un de plus grâce aux bonbons verts.

C'est de l'informatique !

Deux des trois règles de l'exercice sont formulées comme des *instructions conditionnelles* : si une certaine condition est remplie, alors une certaine action est exécutée. De telles instructions conditionnelles sont très souvent utilisées en informatique. Souvent, les mots clés de langue anglaise *if* («si» en anglais) et *then* («alors» en anglais) sont utilisés. L'une des règles est formulée de manière à ce que quelque chose soit répété tant qu'une condition est remplie. On appelle cela une *boucle* pour laquelle on utilise souvent le mot de langue anglaise *while* («tant que» en anglais). De tels boucles peuvent aussi être formulées comme des *boucles itératives* qui déterminent un nombre de répétitions fixe.

On pourrait donc formuler le jeu de Petra de la manière suivante :

```
fixe Tours égal à 3
tant qu'il reste un tour :
  diminue Tours de 1
  tire un bonbon
  tant que le bonbon est vert, alors mets-le dans la coupe et tire un bonbon
  si le bonbon est rouge, alors mets-le dans la coupe
  si le bonbon est jaune, alors mange-le
```

Pour résoudre l'exercice, on doit *analyser* le programme. Dans un cas simple comme celui de ce programme, on pourrait simplement essayer tous les ordres possibles de bonbons. Cela pourrait même être exécuté par un ordinateur de manière automatisée. L'explication de la solution, par contre, se base sur la compréhension des relations qui permet de prouver qu'une certaine solution est vraie sans exécuter le programme. De telles analyses ne peuvent pas toujours être réalisées par un ordinateur, comme l'a montré la *théorie de la calculabilité*. Donald Knuth, un des grands informaticiens du XX^e siècle, l'a exprimé ainsi : «Faites attention aux erreurs dans ce code ; je n'ai fait que démontrer qu'il était correct, je ne l'ai pas essayé.»

Mots clés et sites web

Instruction conditionnelle, boucle, théorie de la calculabilité

- [https://fr.wikipedia.org/wiki/Instruction_conditionnelle_\(programmation\)](https://fr.wikipedia.org/wiki/Instruction_conditionnelle_(programmation))
- https://fr.wikipedia.org/wiki/Structure_de_contr%C3%B4le#Boucles



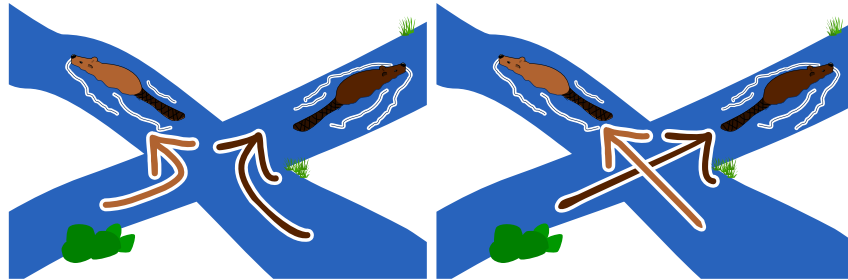
- <https://fr.wikipedia.org/wiki/Calculabilit%C3%A9>
- https://en.wikiquote.org/wiki/Donald_Knuth





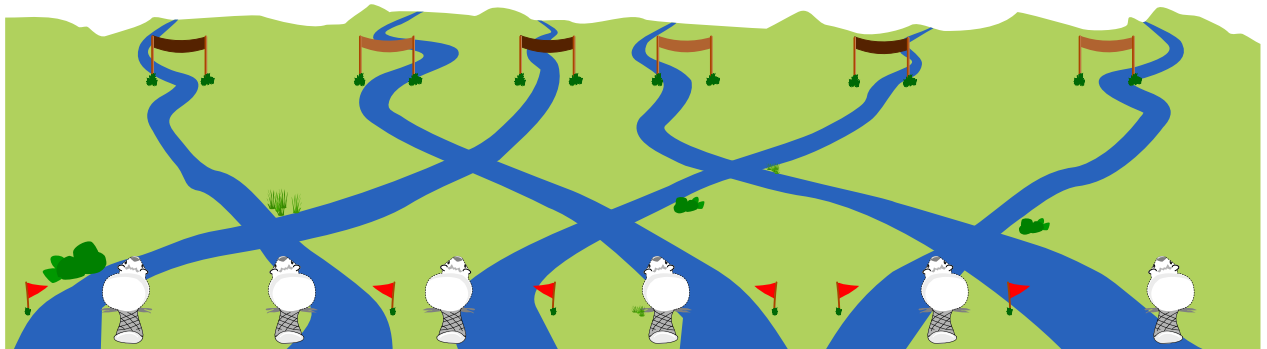
9. Réseau de castors

Trois castors brun clair et trois castors brun foncé nagent dans un système de canaux du bas vers le haut. Deux castors se rencontrent à chaque croisement de deux canaux. Si les deux castors qui se rencontrent sont de couleurs différentes, le castor brun clair continue vers la gauche et le castor brun foncé vers la droite. Sinon, ils continuent simplement chacun dans la même direction.



À la fin, les castors doivent être ordonnés de gauche à droite de la manière suivante : brun foncé, brun clair, brun foncé, brun clair, brun foncé, brun clair.

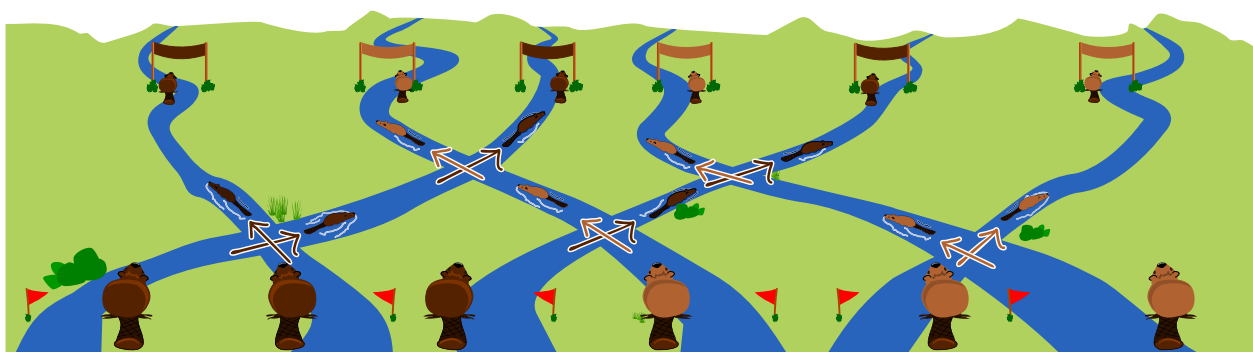
À quelles positions les castors brun clair et brun foncé doivent-ils commencer afin d'arriver dans le bon ordre ?





Solution

Il y a deux bonnes réponses :



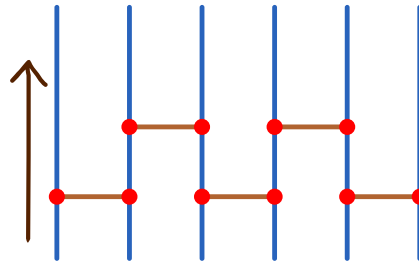
Ce sont également les deux seules réponses justes. En effet, pour que le castor arrivant tout à gauche soit brun foncé, il ne doit pas y avoir de castor brun clair qui passe par le premier croisement à gauche, car celui-ci devrait prendre le canal de gauche. Les deux positions de départ de gauche doivent donc être occupées par des castors brun foncé.

Le même raisonnement vaut pour la position d'arrivée du castor brun clair tout à droite : pour que le castor arrivant tout à droite soit brun clair, deux castors brun clair doivent se rencontrer au premier croisement depuis la droite. Les deux positions de départ de droite doivent donc être occupées par des castors brun clair.

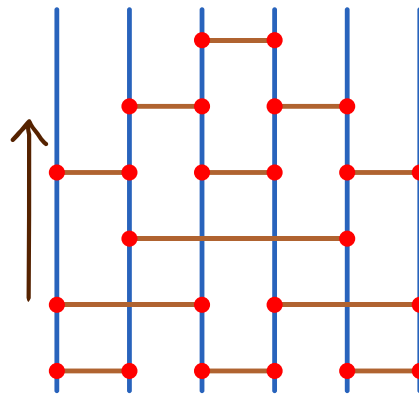
Pour les castors du milieu, cela ne fait pas de différence si le troisième castor brun clair part à gauche et le troisième castor brun foncé part à droite ou l'inverse, car après le premier croisement, le castor brun clair continue de toute façon à gauche et le castor brun foncé à droite.

C'est de l'informatique !

Le système de canaux des castors et la règle déterminant qui nage à gauche et qui nage à droite forment une partie d'un *réseau de tri*. Dans un réseau de tri, des données se déplacent le long d'un fil (les canaux dans cet exercice), et à chaque comparateur (les croisements dans cet exercice), on teste pour décider si les données doivent être échangées ou non. On peut par exemple représenter le castor brun foncé par le chiffre 0 et le castor brun clair par le chiffre 1. Le réseau de tri ressemble alors à cela :



Dans ce diagramme, les lignes bleues représentent les canaux des castors et les points orange reliés par un trait représentent les croisements où les castors changent potentiellement de direction. Ce réseau de tri n'est pas *complet* : il ne va pas forcément disposer tous les castors brun clair à gauche et tous les castors brun foncé à droite. En ce sens, il n'effectue qu'un tri *partiel* des castors. Le réseau de tri suivant est un réseau de tri optimal (complet et minimal) pour cet exercice : en suivant les mêmes règles, ce réseau serait toujours capable de trier tous les castors, quel que soit leur disposition initiale. On peut voir que le réseau de tri ci-dessus est intégré dans le réseau optimal (aux deuxième et troisième lignes depuis le haut) :



Les réseaux de tri sont particulièrement efficaces lorsque les comparaisons peuvent être effectuées parallèlement l'une à l'autre. De tels réseaux de tri optimaux sont difficiles à trouver pour de grands ensembles de données.

Si l'on généralise, on peut aussi se représenter le système de canaux des castors comme le système de câbles d'un réseau informatique comme Internet. Les canaux représentent alors les connexions par câbles directes entre deux routeurs, les intersections. Généralement, des tables de routage fixes permettant de diriger les données vers leur but sont programmées dans de tels routeurs.

Mots clés et sites web

Réseau de tri, réseau informatique, routeur, table de routage

- https://fr.wikipedia.org/wiki/R%C3%A9seau_de_tri
- <http://www.inf.fh-flensburg.de/lang/algorithmen/sortieren/networks/optimal/optimal-sorting-networks.htm>
- <https://www.computernetworkingnotes.com/ccna-study-guide/basic-routing-concepts-and-protocols-explained.html>
- <https://fr.wikipedia.org/wiki/Routage>
- https://fr.wikipedia.org/wiki/Table_de_routage



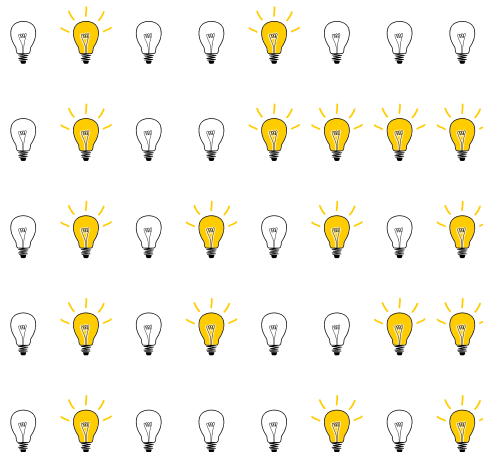


10. Signaux lumineux

Sophie a huit lampes reliées par des interrupteurs et des câbles. Elle peut les utiliser pour envoyer des messages. Pour cela, elle utilise la table de codage suivante, dans laquelle 0 signifie que la lampe correspondante est éteinte (💡) et 1 que la lampe correspondante est allumée (💡) :

A : 01000001	J : 01001010	S : 01010011
B : 01000010	K : 01001011	T : 01010100
C : 01000011	L : 01001100	U : 01010101
D : 01000100	M : 01001101	V : 01010110
E : 01000101	N : 01001110	W : 01010111
F : 01000110	O : 01001111	X : 01011000
G : 01000111	P : 01010000	Y : 01011001
H : 01001000	Q : 01010001	Z : 01011010
I : 01001001	R : 01010010	

Sophie envoie à présent les signaux lumineux suivants :



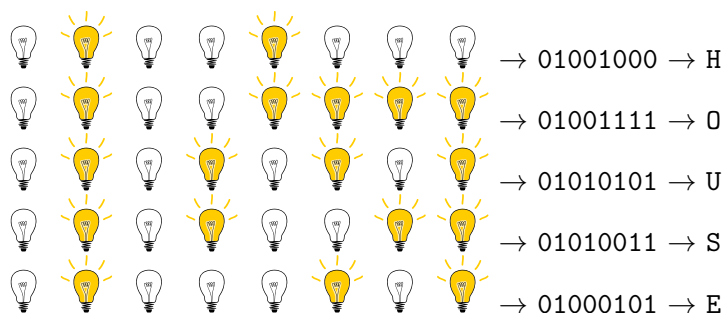
Que signifient les signaux lumineux de Sophie ?

- A) HOUSE
- B) HAPPY
- C) HORSE
- D) HONEY



Solution

Les signaux lumineux signifient :



La bonne réponse est donc le mot A) HOUSE.

On peut par ailleurs trouver cette réponse très facilement : la lettre du milieu est différente dans chaque mot : A) U, B) P, C) R et D) N. Comme le troisième signal lumineux signifie U, seule la réponse A) peut être juste.

C'est de l'informatique !

Le codage de Sophie n'a pas été choisi au hasard. Elle utilise une partie du code appelé ASCII qui a été développé pour l'échange de messages il y a plus de 50 ans. Il est basé sur le principe du code binaire que Gottfried Leibnitz (1646–1716) a déjà décrit en 1679 et 1703 sur la base de systèmes précurseurs indiens et chinois pour la représentation de nombres et le calcul avec ces nombres. Claude Shannon (1916–2001) l'appliqua ensuite lors du développement de l'ordinateur.

Aujourd'hui, les ordinateurs utilisent une version mise à jour de l'ASCII. Comme l'ASCII ne contenait que 95 caractères imprimables (les lettres latines majuscules et minuscules, les chiffres de 0 à 9 ainsi que quelques signes de ponctuation) et que les 33 autres étaient des caractères de contrôle (pour les imprimantes par exemple), on a bientôt eu besoin d'additions pour les accents et les autres alphabets. Cela eut lieu d'abord sous la forme d'un premier code utilisant 8 bits par caractère (ANSI) et ensuite sous la forme du standard Unicode utilisé presque universellement. Les lettres de Sophie sont encore codées exactement de la même manière en UTF-8, la version de l'Unicode la plus répandue.

Le premier groupe de lettres (identique en ASCII, 8 bits et Unicode) est par ailleurs (les caractères de contrôle sont laissés vides, `␣` représente l'espace) :



	000 ...	001 ...	010 ...	011 ...	100 ...	101 ...	110 ...	111 ...
...0000			□	0	@	P	'	p
...0001			!	1	A	Q	a	q
...0010			"	2	B	R	b	r
...0011			#	3	C	S	c	s
...0100			\$	4	D	T	d	t
...0101			%	5	E	U	e	u
...0110			&	6	F	V	f	v
...0111			'	7	G	W	g	w
...1000			(8	H	X	h	x
...1001)	9	I	Y	i	y
...1010			*	:	J	Z	j	z
...1011			+	;	K	[k	{
...1100			,	<	L	\	l	
...1101			-	=	M]	m	}
...1110			.	>	N	^	n	~
...1111			/	?	O	_	o	

Mots clés et sites web

ASCII, Unicode, Codage

- https://fr.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange
- https://fr.wikipedia.org/wiki/Code_binaire
- https://fr.wikipedia.org/wiki/Gottfried_Wilhelm_Leibniz
- https://fr.wikipedia.org/wiki/Claude_Shannon
- <https://fr.wikipedia.org/wiki/Unicode>
- <https://fr.wikipedia.org/wiki/UTF-8>
- <https://www.unicode.org/charts/PDF/U0000.pdf>



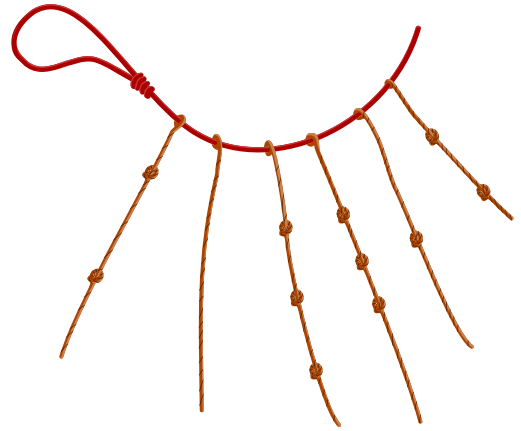


11. Quipu

Les Incas utilisaient à l'époque des nœuds pour la transmission de messages. Plusieurs cordelettes sur lesquelles des nœuds étaient noués étaient attachées à une corde. Ces assemblages de cordelettes appelés « quipus » étaient grands et difficiles à fabriquer.

Imagine qu'il faut développer une version simplifiée des quipus. Les conditions sont :

- Il y a toujours le même nombre de cordelettes attachées à la corde.
- Les cordelettes ne diffèrent que par le nombre de nœuds.
- Une cordelette a 0, 1, 2 ou 3 nœuds.
- L'ordre des cordelettes est déterminé à l'aide d'un nœud sur la corde.
- Il doit pouvoir y avoir 30 quipus discernables pour différents messages.



Quel est le nombre minimum de cordelettes de la version simplifiée des quipus dans ces conditions ?

- A) 2
- B) 3
- C) 4
- D) 5
- E) 8
- F) 10



Solution

La réponse B) 3 est correcte.

Chaque cordelette peut enregistrer une des 4 valeurs différentes (0, 1, 2 ou 3). Avec deux cordelettes, on aurait $4 \cdot 4 = 16$ combinaisons possibles, avec trois cordelettes $4 \cdot 4 \cdot 4 = 64$ combinaisons possibles, et ainsi de suite. Trois cordelettes sont donc suffisantes, plus de cordelettes seraient en contradiction avec la condition stipulant qu'il faut attacher le moins de cordelettes possible à la corde. Comme l'ordre des valeurs est déterminé par le nœud sur la corde, on ne doit pas s'inquiéter du fait que l'on pourrait lire la corde dans un sens ou dans l'autre.

C'est de l'informatique !

Les *quipus* étaient en effet utilisés par les Incas en Amérique du Sud. Des quipus gris étaient utilisés pour la comptabilité et la perception d'impôts. On suppose que jusqu'à 95 syllabes différentes pouvaient être codées à l'aide de cordelettes colorées, permettant une correspondance écrite. Au contraire de la version simplifiée de cet exercice, différentes sortes de nœuds et dans certains cas des sous-cordelettes attachées aux cordelettes étaient utilisées

L'exemple de cet exercice est une version simplifiée. Comme l'ordre est fixé par le nœud sur la corde, les valeurs individuelles (0, 1, 2 ou 3) créent une *notation positionnelle*, dans ce cas un système en base 4. Les notations positionnelles sont très répandues : en règle générale, le système décimal (en base 10) est utilisé, les ordinateurs utilisent le *système binaire* (en base 2). Au temps des premiers ordinateurs, il y a eu des essais de construction d'ordinateurs basés sur le *système ternaire* (en base 3, représentés dans ce cas par -1 , 0 et $+1$). En utilisant un système en base b avec n positions, on peut représenter exactement b^n valeurs différentes. Un octet (8 bits qui peuvent valoir chacun 0 ou 1) peut de cette manière enregistrer $2^8 = 256$ valeurs différentes (de 0 à 255), le quipu de cet exercice $4^3 = 64$ valeurs différentes

Les Incas n'auraient eu besoin que d'une seule cordelette pour enregistrer les valeurs de 1 à 30. Ils utilisaient également un système décimal comme nous pour l'écriture des nombres, simplement en utilisant différents nœuds sur une cordelette. La position des unités aurait été codée entre autres avec des demi-nœuds et la position des dizaines avec un demi-nœud auquel un nombre de tours correspondants était ajouté. Cependant, ils auraient eu besoin d'au moins 4 nœuds et de plusieurs sortes de nœuds.

Mots clés et sites web

Quipu, notation positionnelle

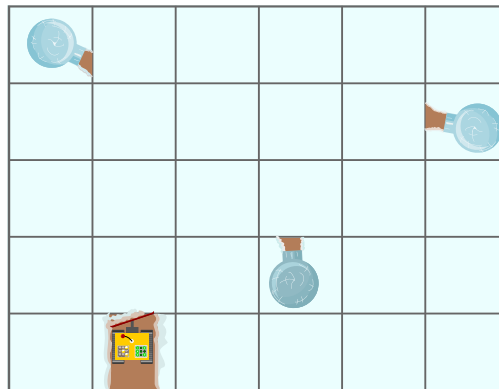
- <https://fr.wikipedia.org/wiki/Quipu>
- https://fr.wikipedia.org/wiki/N%C5%93ud_double
- https://en.wikipedia.org/wiki/Stopper_knot
- https://fr.wikipedia.org/wiki/Notation_positionnelle
- https://fr.wikipedia.org/wiki/Ordinateur_ternaire



12. Tempête de neige

Il y a des congères partout après une forte tempête de neige et les habitants des trois iglous sont isolés. Les habitants peuvent dégager des chemins à l'aide de leur chasse-neige télécommandé. Cela fonctionne ainsi :

- Le chasse-neige a besoin de 4 minutes pour passer d'une case à une case voisine en la dégageant.
- Le chasse-neige a besoin d'une minute pour passer d'une case déjà déneigée à une case voisine.
- Les case voisines ne sont que les cases qui sont situées directement en dessus, en dessous, à gauche ou à droite d'une case sur la carte. La chasse-neige ne peut donc pas rouler en diagonale.
- Dès que la case devant l'entrée d'un iglou est dégagée, les habitants de l'iglou peuvent en dégager l'entrée avec une pelle et ne sont plus isolés.

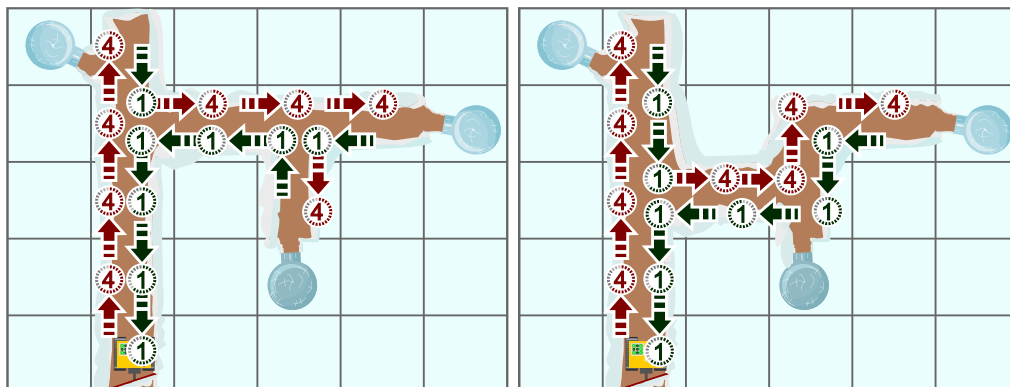


Dans le cas idéal, de combien de minutes le chasse-neige a-t-il besoin pour libérer tous les iglous de leur isolement et retourner à sa case de départ ?



Solution

La bonne réponse est 40. Les illustrations suivantes montrent les deux chemins optimaux pour le chasse-neige :



Pourquoi est-ce que ça ne va pas plus rapidement ? Pour atteindre l'iglou en haut à gauche, il faut dégager quatre cases. Cela prend 16 minutes. Pour atteindre l'iglou à droite, trois cases de plus doivent être dégagées. Cela prend 12 minutes de plus. Pour atteindre l'iglou du bas, une case doit encore être dégagée, car il faut soit dégager un petit chemin jusqu'au chemin transversal, soit ajouter un virage au chemin transversal. Cela prend encore 4 minutes. Pour que le chasse-neige retourne à la case départ, il doit encore parcourir quatre cases vers le bas et trois cases vers la gauche. Cela dure encore 7 minutes. Le détour par le petit chemin ou le virage dans le chemin transversal dure encore une minute supplémentaire. Il a donc besoin de 40 minutes en tout.

Si le chasse-neige dégagait le chemin plus vite, ce serait éventuellement plus efficace s'il rentrait de l'iglou du bas en passant par la case enneigée à gauche en la dégagant. Mais il a besoin de 4 minutes pour la dégager et d'une minute pour passer sur la case déjà déneigée, donc de 5 minutes, alors que le détour par les cases déjà déneigées ne lui prend que 4 minutes.

C'est de l'informatique !

Il s'agit dans cet exercice de trouver un réseau de chemins qui relie tous les endroits (les iglous et la case de départ) à un coût minimal (le temps dont le chasse-neige a besoin). De tels réseaux ne sont pas forcément composés des chemins les plus courts entre tous les endroits, mais les coûts nécessaires à la réalisation du réseau doivent être aussi bas que possible. On appelle de tels réseaux des *arbres de Steiner*. Ils sont utilisés, par exemple, pour la fabrication des cartes mères des ordinateurs ou la construction de réseaux ferroviaires peu exploités pour le transport de marchandises. La recherche d'arbres de Steiner est un problème d'optimisation difficile et qui prend beaucoup de temps en informatique, ce qui fait que l'on utilise souvent des algorithmes qui trouvent des solutions suffisamment bonnes, mais pas forcément la meilleure solution.

Dans cet exercice, les coûts sont calculés de manière spéciale, car il n'y a pas seulement un coût fixe pour l'élaboration d'un chemin (les 4 minutes pour dégager une case), mais également un coût pour le déplacement de la machine sur les cases déneigées. Cet exercice est donc une généralisation du problème de l'arbre de Steiner.

Mots clés et sites web

Problème de l'arbre de Steiner

— https://fr.wikipedia.org/wiki/Probl%C3%A8me_de_l%27arbre_de_Steiner



13. Quel bonheur que les arbres !

Sergio a écrit une chanson qui décrit comment plusieurs choses différentes peuvent se former sur un arbre. Voici un couplet :

Quel bonheur que les arbres !
Sur un arbre poussent des feuilles,
Sur un arbre poussent des fleurs,
Les fleurs donnent des fruits,
Avec des feuilles et des fleurs, je peux tresser des couronnes.

C'était important pour Sergio de n'utiliser après le premier vers que des objets déjà mentionnés auparavant.

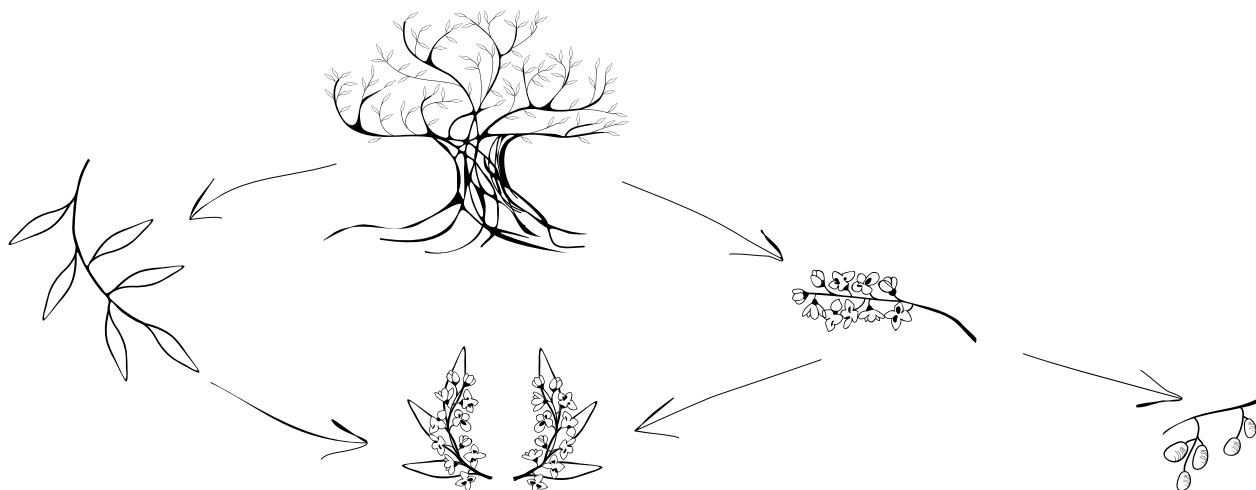
Lequel des couplets suivants est faux d'après Sergio ?

- A) Quel bonheur que les arbres !
Sur un arbre poussent des fleurs,
Sur un arbre poussent des feuilles,
Avec des feuilles et des fleurs, je peux tresser des couronnes,
Les fleurs donnent des fruits.
- B) Quel bonheur que les arbres !
Sur un arbre poussent des fleurs,
Sur un arbre poussent des feuilles,
Les fleurs donnent des fruits,
Avec des feuilles et des fleurs, je peux tresser des couronnes.
- C) Quel bonheur que les arbres !
Sur un arbre poussent des feuilles,
Les fleurs donnent des fruits,
Sur un arbre poussent des fleurs,
Avec des feuilles et des fleurs, je peux tresser des couronnes.
- D) Quel bonheur que les arbres !
Sur un arbre poussent des fleurs,
Les fleurs donnent des fruits,
Sur un arbre poussent des feuilles,
Avec des feuilles et des fleurs, je peux tresser des couronnes.
- E) Quel bonheur que les arbres !
Sur un arbre poussent des feuilles,
Sur un arbre poussent des fleurs,
Avec des feuilles et des fleurs, je peux tresser des couronnes,
Les fleurs donnent des fruits.



Solution

On peut représenter les dépendances des objets «arbre», «feuilles», «fleurs», «couronnes» et «fruits» à l'aide d'un graphe sur lequel une flèche signifie qu'un objet est nécessaire à la formation de l'autre :



D'après ce graphe, les fleurs doivent être mentionnées avant les fruits, et les feuilles et les fleurs avant les couronnes.

La séquence des objets dans les différentes réponses est la suivante :

- A) Arbre, fleurs, feuilles, couronnes, fruits
- B) Arbre, fleurs, feuilles, fruits, couronnes
- C) Arbre, feuilles, *fruits*, *fleurs*, couronnes
- D) Arbre, fleurs, fruits, feuilles, couronnes
- E) Arbre, feuilles, fleurs, couronnes, fruits

Dans la réponse C), les fruits sont mentionnés avant les fleurs (en italique plus haut), ce qui est une contradiction car il faut des fleurs pour faire des fruits. Tous les autres vers respectent les conditions.

C'est de l'informatique !

En 1974, le musicien italien Sergio Endrigo (1933–2005) a écrit la chanson pour enfants « Ci vuole un fiore » d'après un texte de Gianni Rodari (1920–1980). Dans cette chanson, il chante que l'on a d'abord besoin de bois pour avoir une table, d'un arbre pour avoir du bois, d'une graine pour avoir un arbre, d'un fruit pour avoir une graine et d'une fleur pour avoir un fruit. Il décrit aussi que pour avoir une fleur, on a également besoin d'une fleur en passant par une branche, un arbre, une forêt, une montagne et la terre. Il finit par dire qu'on a en fin de compte besoin d'une fleur pour tout.

La précedence d'un objet sur un autre peut être décrite à l'aide d'un *graphe orienté*. Un tel graphe est dessiné dans l'explication de la réponse. C'est un *graphe orienté acyclique* qui décrit un *ensemble de séquences admissibles*. Si l'on veut un nœud (l'un des objets), on doit déjà avoir tous les objets pointant vers lui. Le même chose est vraie pour ces objets-là, ce qui fait que l'on doit remonter de manière *réursive* jusqu'à ce que l'on arrive à un objet sur lequel rien ne pointe. On peut utiliser cet objet comme objet de départ

On ne peut d'ailleurs pas décrire la chanson de Sergio Endrigo à l'aide d'un graphe orienté acyclique. Dans la deuxième partie de la chanson décrite plus haut, il chante que l'on a en fin de compte besoin d'une fleur pour avoir une fleur. C'est en contradiction avec le fait que le graphe doit être acyclique



et ne peut donc pas contenir de circuit. Avec cette rupture de logique, il rend sa déclaration encore plus claire: «Ci vuole un fiore» — «Nous avons besoin d'une fleur»!

Mots clés et sites web

Graphe orienté acyclique, tri topologique

- <https://www.filastrocche.it/contenuti/ci-vuole-un-fiore/>
- <https://www.youtube.com/watch?v=9ht4tIot8XY>
- https://en.wikipedia.org/wiki/Precedence_graph
- https://fr.wikipedia.org/wiki/Tri_topologique

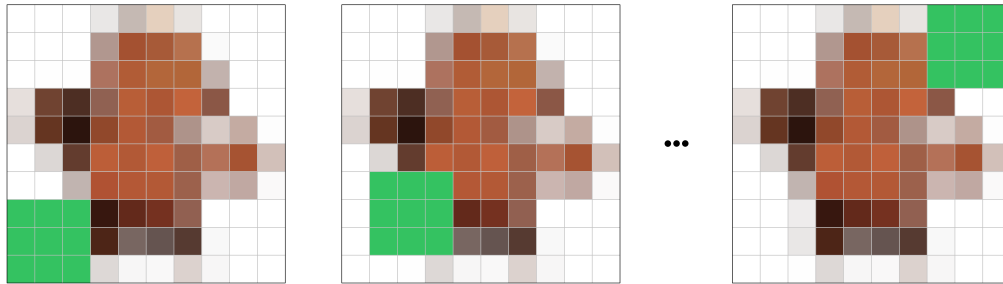




14. Compression vidéo

Les vidéos occupent beaucoup d'espace de stockage. Pourtant, deux images fixes consécutives se ressemblent souvent beaucoup.

La vidéo suivante a une taille de 10×10 points. Le carré vert dans le coin en bas à gauche fait 3×3 points. Il se déplace d'image fixe à image fixe d'un point vers la droite et d'un point vers le haut à chaque image jusqu'à ce qu'il arrive dans le coin en haut à droite.



Pour économiser de l'espace de stockage, à partir de la deuxième image, seuls les points qui ont changé sont enregistrés.

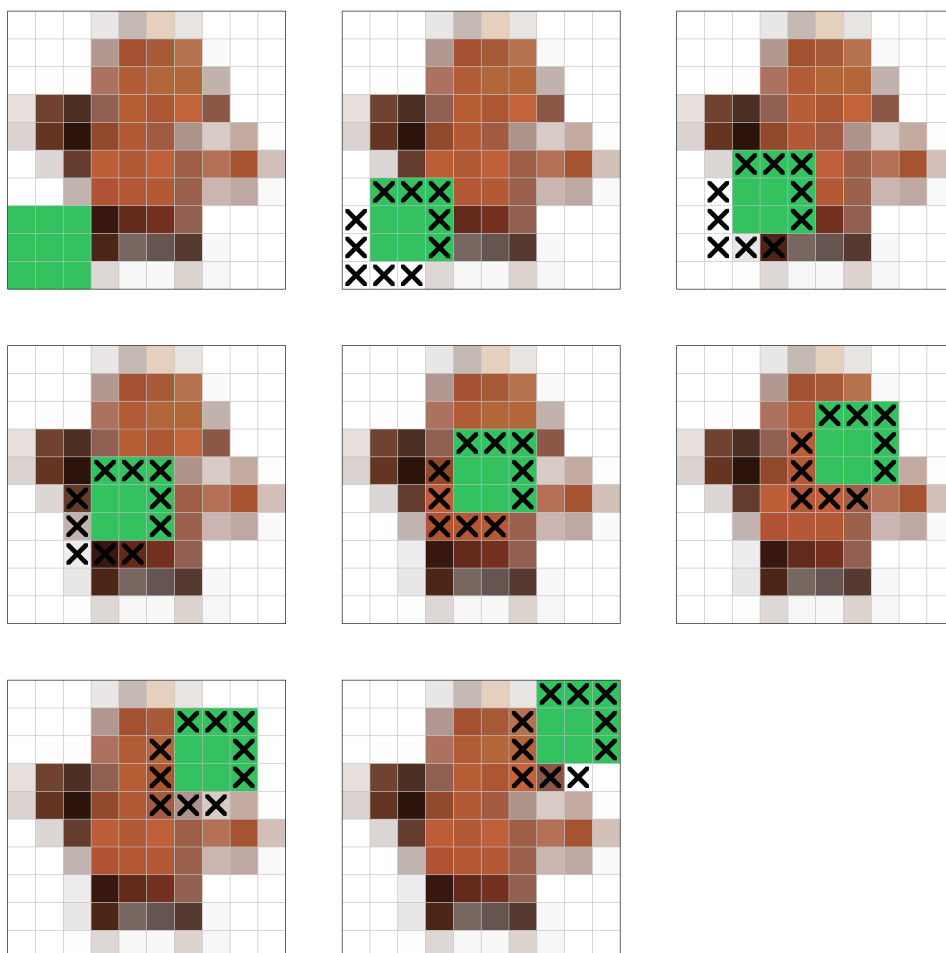
Combien de points doivent être enregistrés pour toute la vidéo ?

- | | | |
|--------|--------|---------|
| A) 100 | D) 170 | G) 800 |
| B) 135 | E) 180 | H) 1000 |
| C) 140 | F) 700 | |



Solution

Les images fixes individuelles ressemblent à cela lorsque l'on marque les points ayant changé :



On remarque tout d'abord que la première image contient $10 \cdot 10 = 100$ points. Pour chacune des images suivantes, seuls les points qui ont changé doivent être enregistrés. Il s'agit des cinq points en bas à gauche du carré qui sont remplacés par les points de l'arrière-plan et des cinq points en haut à droite sur le carré qui représentent la nouvelle position du carré. 10 points changent donc par image fixe. Le carré a besoin de 7 images fixes supplémentaires pour se déplacer d'en bas à gauche vers en haut à droite, il faut donc ajouter $10 \cdot 7 = 70$ points pour les points qui changent aux 100 points de la première image. La bonne réponse est donc D) 170 points.

C'est de l'informatique !

Comme décrit dans l'exercice, la compression vidéo joue un rôle important de nos jours. La méthode décrite ici n'est qu'une des approches permettant de compresser des vidéos. Une autre approche consiste à enlever certaines informations qu'une personne ne perçoit habituellement pas. Le format d'image JPEG utilise de telles relations. Sur des images compressées spécialement fortement, on peut le reconnaître à la formation de blocs, car la différence entre les couleurs du bloc a été interprétée à tort comme imperceptible. Une autre possibilité est de réduire la dimension de l'espace de couleur. C'est sur ces idées qu'est basé le standard MPEG. Comme dans cet exercice, il fait la différence entre différents types d'images fixes. Un type d'images fixes (appelées images «intra») représente une image fixe complète (similairement à notre première image fixe). Un autre type d'images fixes



est basé sur la différence avec l'image fixe précédente (images «P», comme nos autres images fixes) ou même sur la différence avec l'image fixe suivante en plus (images «D», ne sont pas utilisées dans cet exercice). Pour minimiser l'utilisation de mémoire tampon et pour pouvoir «retrouver le fil» en cas d'erreurs de transmission, des images intra sont insérées à intervalles réguliers. Sur des vidéos fortement compressées, on peut reconnaître les images P et D lorsque qu'un arrière-plan sombre «saute» tout à coup alors que la scène n'a changé que lentement pendant un certain temps.

Le besoin en espace de stockage n'est pas vraiment aussi petit que suggéré dans l'exercice : en plus des valeurs des couleurs, la position du pixel qui a changé doit aussi être enregistrée. Cela multiplie peut-être par deux l'espace de stockage nécessaire à un pixel changé. Même ainsi, 240 unités de stockage comparées à 800 unités de stockage représentent une économie de place impressionnante, d'autant plus que la méthode décrite dans l'exercice est, contrairement au MPEG, une méthode sans perte !

Mots clés et sites web

Compression vidéo

- https://fr.wikipedia.org/wiki/Compression_vid%C3%A9o
- <https://fr.wikipedia.org/wiki/JPEG>
- <https://fr.wikipedia.org/wiki/MPEG-1>



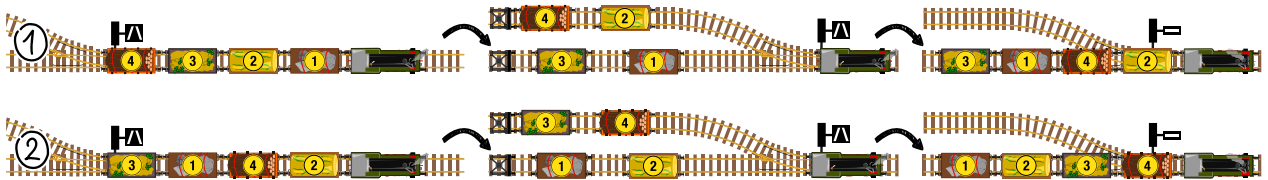


15. Gare de triage

Un train de marchandises doit amener des wagons de marchandises individuels sur des voies de raccordement le long de la ligne principale. Pour économiser du temps et éviter de manœuvrer sur la ligne principale, les wagons de marchandises doivent être classés d'après leur numéro de façon à ce que le wagon portant le numéro 1 soit tout à gauche.

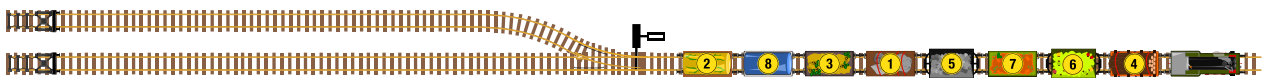
Dans la gare de triage, il y a une butte sur laquelle les wagons de marchandises sont refoulés vers le bas de gauche à droite. Sur la butte, on décide pour chaque wagon sur laquelle des deux voies de garage il est orienté. Ensuite, la locomotive tracte les wagons hors des voies de garage : d'abord tous ceux se trouvant sur une voie, ensuite tous ceux se trouvant sur l'autre voie de garage. On considère ces actions comme une étape de tri.

Par exemple, lorsque quatre wagons de marchandises doivent être triés, deux étapes de tri suffisent (étape ① et étape ②) :



Ce n'est pas possible de trier les wagons en une seule étape de tri.

Si les wagons de marchandises sont dans l'ordre 2 – 8 – 3 – 1 – 5 – 7 – 6 – 4, de combien d'étapes de tri a-t-on besoin au minimum pour trier le train de marchandises ?



- A) 3
- B) 4
- C) 5
- D) 6
- E) 7
- F) 8



Solution

La bonne réponse est que l'on a besoin de A) 3 étapes de tri.

Bien sûr que l'on peut utiliser plusieurs méthodes pour trier le train de marchandises, mais l'une des meilleures méthodes est de commencer par refouler les wagons de marchandises 1, 3, 5 et 7 sur la voie du haut et les wagons 2, 4, 6, et 8 sur la voie du bas, puis de tracter d'abord les wagons de la voie du bas puis ceux de la voie du haut :



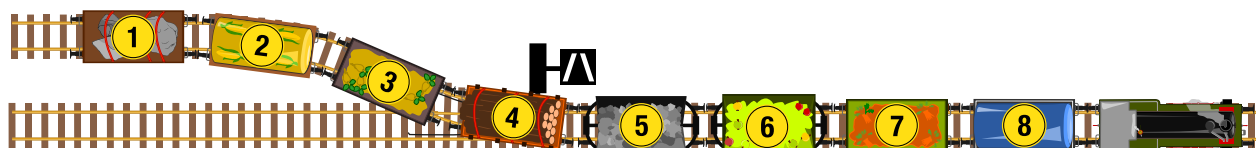
Ainsi, pour chaque paire (1 et 2, 3 et 4, 5 et 6, 7 et 8) de wagons de marchandises, le wagon portant le plus petit numéro se trouve toujours à gauche de celui portant le plus grand numéro.

Ensuite, c'est raisonnable de refouler les wagons de marchandises 1, 2, 5 et 6 sur la voie du haut et les wagons 3, 4, 7 et 8 sur la voie du bas, puis de tracter d'abord les wagons de la voie du bas puis ceux de celle du haut :



On n'a ainsi pas changé l'ordre des paires obtenues auparavant, car chaque paire a été refoulée sur la même voie. On a en plus mis les wagons 1 à 4 et 5 à 8 dans le bon ordre relatif les uns aux autres, mais les deux groupes ne sont pas encore mélangés.

Finalement, il ne faut plus que refouler les wagons de marchandises 1 à 4 sur la voie du haut et les wagons 5 à 8 sur la voie du bas, puis de tracter d'abord les wagons de marchandise de la voie du bas, puis de la voie du haut :



L'ordre dans chaque groupe n'a pas été modifié, car tous les wagons de marchandises du groupe 1 à 4 ont été refoulés sur une voie et tous les wagons du groupe 5 à 8 sur l'autre voie. Maintenant, les deux groupes sont composés de wagons de marchandises ordonnés et tous les wagons de l'un des deux groupes ont des numéros plus petits que tous les wagons de l'autre groupe.

On ne peut pas trier les huit wagons plus rapidement. Une preuve complète de cela serait trop longue pour cet exercice, mais l'idée de base est la suivante: lors d'une étape de tri, on peut changer l'ordre d'un sous-ensemble relativement aux autres sous-ensembles, mais pas l'ordre dans les sous-ensembles mêmes. On ne peut donc trier que deux wagons de marchandises dans un ordre défavorable lors de la première étape de tri. Chaque étape de tri suivante double le nombre de wagons ordonnés défavorablement que l'on peut trier. Les huit wagons de marchandises de cet exercice sont choisis de manière à être ordonnés défavorablement, donc deux étapes de tri ne suffisent pas.

C'est de l'informatique !

Les cheminots du monde entier doivent résoudre de tels problèmes quotidiennement, car le tri de wagons de marchandises est une tâche qui demande beaucoup de temps et de travail: chaque fois,



les wagons doivent être connectés et déconnectés, ce qui est encore un travail manuel. Cela prend du temps et bloque la ligne principale, surtout lorsque plusieurs wagons de marchandises doivent être sécurisés et déconnectés. C'est pour cela que des cheminots ont très tôt développé de grandes gares de triage avec beaucoup de voies de garage. En Suisse, il y a des gares de triage à MuttENZ près de Bâle, à Buchs (Saint-Gall), entre Spreitenbach et Dietikon près de Zurich, à Denges près de Lausanne et à Chiasso. Dans cet exercice, la gare de triage n'a que deux voies de garage, un défi pour les grands trains de marchandises mais une situation typique pour les lignes ferroviaires secondaires, en particulier pour les lignes de chemin de fer à voie étroite qui n'ont pas de connexion directe aux grandes entreprises ferroviaires.

L'informatique peut être très utile pour trier efficacement les trains de marchandise. Dans ce cas, le principe consistant à résoudre encore et encore le même problème simplifie fortement l'exercice : une méthode connue en informatique sous le nom de «diviser pour régner» (*divide & conquer* en anglais). Dans ce cas, on trie d'abord les paires de wagons de marchandises, puis des groupes de quatre wagons, puis le groupe de huit wagons.

Les voies de garage pour les wagons de marchandises fonctionnent comme le type de données abstrait *pile*, qui est beaucoup utilisé en informatique. Les seules opérations autorisées sont : *dépiler* (enlever l'élément du dessus, *pop* en anglais) et *empiler* (ajouter un élément sur la pile, *push* en anglais). Parfois, on peut aussi *voir l'élément de tête* (*top* en anglais) et *vérifier si la pile est vide* (*empty* en anglais).

Mots clés et sites web

Diviser pour régner (Divide & Conquer), pile

- https://fr.wikipedia.org/wiki/Gare_de_triage
- [https://fr.wikipedia.org/wiki/Diviser_pour_r%C3%A9gner_\(informatique\)](https://fr.wikipedia.org/wiki/Diviser_pour_r%C3%A9gner_(informatique))
- [https://fr.wikipedia.org/wiki/Pile_\(informatique\)](https://fr.wikipedia.org/wiki/Pile_(informatique))



A. Auteurs des exercices

 Tony René Andersen	 Thomas Ioannou	 Zsuzsa Pluhár
 Michelle Barnett	 Takeharu Ishizuka	 Sergei Pozdniakov
 Michael Barot	 Anna Laura John	 Nol Premasathian
 Wilfried Baumann	 Mile Jovanov	 J.P. Pretti
 Jan Berki	 Ungyeol Jung	 Milan Rajković
 Špela Cerar	 Injoo Kim	 Chris Roffey
 Mony Chanroath	 Jihye Kim	 Andrea Schrijvers
 Marios Choudary	 Mária Kiss	 Eljakim Schrijvers
 Anton Chukhnov	 Sophie Koh	 Humberto Sermeno
 Kris Coolsaet	 Dennis Komm	 Daigo Shirai
 Allira Crowe	 Bohdan Kudrenko	 Jacqueline Staub
 Christian Datzko	 Regula Lacher	 Nikolaos Stratis
 Maria Suyana Datzko	 Inggriani Liem	 Maciej M. Sysło
 Susanne Datzko	 Judith Lin	 Bundit Thanasopon
 Guillaume de Moffarts	 Violetta Lonati	 Nicole Trachsler
 Lanping Deng	 Mattia Monga	 Jiří Vaníček
 Gerald Futschek	 Samart Moodleah	 Florentina Voboril
 Sonali Gogate	 Madhavan Mukund	 Michael Weigend
 Arnheiður Guðmundsdóttir	 Tom Naughton	 Jing-Jing Yang
 Martin Guggisberg	 Pia Niemelä	 Xing Yang
 Juraj Hromkovič	 Tomohiro Nishida	 Khairul A. Mohamad Zaki
 Alisher Ikramov	 Assylkan Omashev	



B. Sponsoring : Concours 2019


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

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


**bischof
berger** <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
Software Services <http://www.bbv.ch/>


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


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


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



<https://educatec.ch/>
educaTEC



<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

Scuola universitaria professionale
della Svizzera italiana



<http://www.supsi.ch/home/supsi.html>
La Scuola universitaria professionale della Svizzera italiana
(SUPSI)



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



C. Offres ultérieures

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!E

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

Devenez vous aussi membre de la SSIE

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

et soutenez le Castor Informatique par votre adhésion

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

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