Le base-ball multicolore :

pensée algorithmique et raisonnement

Groupe algorithmique de l'IREM de Grenoble
{Anne.Rasse,Jean-Marc.Vincent,Benjamin.Wack}@imag.fr
{Maryline.Bruel,Nathalie.Brasset}@ac-grenoble.fr

11/06/19

Plan

Remarque : Cliquez sur les références* pour un accès aux ressources en ligne.

Algorithme

Une première définition

"un algorithme est un procédé qui permet de résoudre un problème, sans avoir besoin d'inventer une solution à chaque fois. " (Abiteboul et Dowek, 2017, p11)

L'homme a toujours utilisé et transmis des algorithmes.
En informatique et en mathématiques les algorithmes portent sur des symboles.
Exemles d'algorithmes en mathématiques :

- Algorithme d'Euclide 300 av JC
  Site hist-math.fr : Le boeuf en Daube*
- Muhammad Musa al-Khwarizmi IX^{e} siècle Kitab Al jabr w'al mouqabala
  Site hist-math.fr : Dixit Algorithmi*

Comment décrire un algorithme ?

Seulement 4 ingrédients suffisent à écrire tous les algorithmes symboliques :

- l'affectation
- la séquence
- la boucle
- le test

La manière d'exprimer un algorithme dépent de l'interlocuteur et de son niveau de connaissance.

Ceci permet de définir des actions élémentaires.

Les ingrédients d'un algorithme peuvent être d'autres algorithmes qui sont alors considérés comme de nouvelles "primitives".

Qu'est qu'un algorithme ?

Une méthode systématique exprimée comme une composition d'actions élémentaires permettant d'exprimer en un nombre fini d'étapes une solution à un problème.

Algorithmes un concept clé en informatique

L'informatique est fondée sur 4 concepts :

- algorithme
- machine
- langage
- information

(Dowek, 2011)

Informatique débranchée

Transmettre des notions fondamentales de l'informatique sans recours à l'ordinateur.

"si on peut passer des heures à cliquer sur une souris sans rien apprendre d'informatique, on peut aussi apprendre beaucoup d'informatique sans toucher une souris." (Di Cosmo, 2015)

Development of Computational Thinking Skills through Unplugged Activities in Primary School (Brackman, Roman-Gonzales and all,2017)

- Computer Science Unpplugged (Tim Bell,Ian H. Witten et Mike Fellows,VF équipe interstices, 2014)*
- Collectif, Informatique débranchée, Tangente éducation numéro 42-43, 2017
- IREM Grenoble*
- IREM Clermont-Ferrand*
- Page médiation scientifique de Marie Duflot*
- Page médiation scientifique de Martin Quinson*

Le base-ball multicolore

Origine CSUnplugged* : Le jeu de l'orange, acheminement et blocage dans les réseaux (Bell, Witten et Fellows, 2005)

Objectifs :

- résolution des problèmes en coopération
- raisonnement logique

IREM Grenoble*

Objectifs :

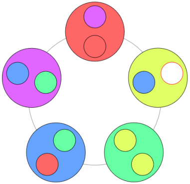
- **questionnement autour des solutions : existence, construction**, complexité,
- **raisonnements algorithmiques et logiques**

Page médiation scientifique de Marie Duflot*

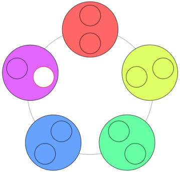
Page médiation scientifique de Martin Quinson*

But du jeu

On dispose de cinq bases de couleurs différentes, et deux pions de la même couleur associés à chaque base. Cependant une base (la violette sur l'exemple) ne possède qu'un seul pion.



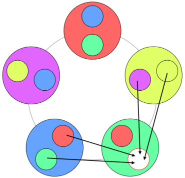
Le but du jeu est de déplacer les pions afin d'amener chaque pion sur la base correspondant à sa couleur.



Règles à respecter

Pour déplacer les pions il faut respecter trois contraintes :

- les bases sont disposées en cercle, et un pion ne peut se déplacer que vers les deux bases voisines (il ne peut pas traverser le terrain)
- on ne peut déplacer qu'un pion à la fois
- chaque base a deux places, et un pion ne peut se déplacer vers une base que si elle possède une place libre



Accès simulation du base-ball*

Jeu libre

Placez les pions au hasard sur les bases, puis cherchez à résoudre cette instance du problème.
Parvenez-vous à dégager un algorithme ou au moins quelques principes généraux ?

Jeu libre

Placez les pions au hasard sur les bases, puis cherchez à résoudre cette instance du problème.
Parvenez-vous à dégager un algorithme ou au moins quelques principes généraux ?

Rappel des contraintes :

- les bases sont disposées en cercle, et un pion ne peut se déplacer que vers les deux bases voisines (il ne peut pas traverser le terrain)
- on ne peut déplacer qu'un pion à la fois
- chaque base a deux places, et un pion ne peut se déplacer vers une base que si elle possède une place libre

Réduisons l'espace de recherche

On peut rapidement tourner en rond :

- 4 mouvements légaux à chaque étape
- on peut « revenir en arrière » à volonté

Une contrainte supplémentaire pour mettre au point une méthode

On fixe un sens de rotation, par exemple en sens horaire.

Réduisons l'espace de recherche

On peut rapidement tourner en rond :

- 4 mouvements légaux à chaque étape
- on peut « revenir en arrière » à volonté

Une contrainte supplémentaire pour mettre au point une méthode

On fixe un sens de rotation, par exemple en sens horaire.

Il reste à choisir lequel déplacer parmi 2 pions : il semble logique de commencer par celui qui est le plus loin de sa destination.

Mélangez bien à nouveau les pions, puis essayez cette stratégie.

Correction de la méthode « tournante »

- Il faut parfois déloger un pion placé sur sa base pour libérer le chemin.
- Malgré cela, ça fonctionne... sur (presque) toutes les situations essayées.
- Peut-on en déduire que ça marche à tous les coups ?

Correction de la méthode « tournante »

- Il faut parfois déloger un pion placé sur sa base pour libérer le chemin.
- Malgré cela, ça fonctionne... sur (presque) toutes les situations essayées.
- Peut-on en déduire que ça marche à tous les coups ?

Contre-exemple

- partir du problème résolu
- échanger deux pions de couleurs distinctes
- la méthode tournante entre alors dans une « boucle infinie » (algorithme non terminant)

Correction de la méthode « tournante »

- Il faut parfois déloger un pion placé sur sa base pour libérer le chemin.
- Malgré cela, ça fonctionne... sur (presque) toutes les situations essayées.
- Peut-on en déduire que ça marche à tous les coups ?

Contre-exemple

- partir du problème résolu
- échanger deux pions de couleurs distinctes
- la méthode tournante entre alors dans une « boucle infinie » (algorithme non terminant)

L'astuce de l'animateur

Si les pions sont « bien mélangés » (chacun est placé sur une base différente de sa couleur), alors l'algorithme tournant fonctionne dans plus 97% des cas pour 4, 5 ou 6 bases.

Généralisons

On considère maintenant n bases, avec chacune 2 pions associés (sauf une qui n'a que 1 pion).
Peut-on déterminer un algorithme (qui fonctionne pour toutes les instances) ?

Raisonnement par récurrence

Si je sais :

- placer deux pions sur leur base
- résoudre le problème pour les $n-1$ bases restantes

alors je saurai résoudre le problème à n bases.

Quelques précautions

Il faut rendre le problème à $n-1$ bases comparable au problème de taille n :

- on « coupe » le cercle en un point et on considère plutôt une ligne
- on commence par une des bases extrêmes (mais pas celle qui n'a qu'un pion)

Au fait, où est l'initialisation ?

Placer deux pions sur leur base

On néglige les couleurs des autres pions :

1. choisir un des 2 pions qu'on souhaite ranger
2. « amener le trou » entre ce pion et sa base
3. l'approcher de cette base et recommencer autant de fois que nécessaire
4. puis idem avec l'autre pion

Placer deux pions sur leur base

On néglige les couleurs des autres pions :

1. choisir un des 2 pions qu'on souhaite ranger
2. « amener le trou » entre ce pion et sa base
3. l'approcher de cette base et recommencer autant de fois que nécessaire
4. puis idem avec l'autre pion

Terminaison

- L'étape 2 prend au maximum $n-2$ opérations.
- Chaque itération de l'étape 3 prend 3 opérations, et sera répétée au plus $n-1$ fois.

Soit au total $4n-5$ opérations pour chaque pion de rang n.

L'algorithme se termine en moins de $4n^2$ opérations.

D'autres approches possibles

Démontrer (constructivement) qu'on sait :

- placer n'importe quel pion dans le trou
  OU
- échanger deux pions sur deux bases voisines

en laissant le reste des pions inchangé

puis résoudre le problème à l'aide de cette « nouvelle » primitive.

Placer n'importe quel pion dans le trou

1. choisir le pion que l'on souhaite échanger et le sens de trajet le plus court du pion au trou On note baseP la base du pion et baseT la base du trou dans la situation de départ.
2. décaller toute la ligne du pion en direction de baseT afin que le trou se retrouve sur la baseP
3. approcher le pion de baseP (décaller 2 pions vers baseP et le pion vers baseT) et recommencer autant de fois que nécessaire
4. décaller les pions en direction de baseP afin qu'ils retrouvent leur place initiale

Placer n'importe quel pion dans le trou

1. choisir le pion que l'on souhaite échanger et le sens de trajet le plus court du pion au trou On note baseP la base du pion et baseT la base du trou dans la situation de départ.
2. décaller toute la ligne du pion en direction de baseT afin que le trou se retrouve sur la baseP
3. approcher le pion de baseP (décaller 2 pions vers baseP et le pion vers baseT) et recommencer autant de fois que nécessaire
4. décaller les pions en direction de baseP afin qu'ils retrouvent leur place initiale

Terminaison

- L'étape 2 prend au maximum $i - 1$ opérations et l'étape 4 $i - 2$ opérations
- Chaque itération de l'étape 3 prend 3 opérations, et sera répétée au plus $i - 1$ fois.

Soit au total $5i - 6$ opérations pour chaque chemin de longueur i bases entre baseP et baseT.

Algorithme utilisant la primitive **placer n'importe quel pion dans le trou**

Répéter pour chaque pion :

1. Echanger le pion avec la couleur de la base du trou.
   Si besoin échanger un pion qui n'est pas encore sur sa base.

Terminaison suite

Placer n'importe quel pion dans un trou en laissant le reste des pions inchangé nécessite $5i - 6$ opérations pour chaque chemin de longueur i bases entre baseP et baseT. La longueur d'un chemin est majorée par $E(n/2) + 1$

L'algorithme se termine en moins de $5n^2$ opérations.

Echanger deux pions *a* et *b* sur deux bases voisines

On considère les deux bases voisines

- 1er cas : le trou se trouve sur l'une des deux bases. On peut aisément échanger les pions *a* et *b* en 2 opérations.
- 2e cas : le trou ne se trouve pas sur l'une des deux bases.
  1. « déplacer le trou »sur une base voisine des deux bases considérées
  2. déplacer un pion autre que *a* ou *b* afin de se ramener au 1er cas
  3. 1er cas
  4. remettre le pion de l'étape 2 à sa place
  5. remettre les pions déplacés à l'étape 1 à leur place en « ramenant le trou »à sa position initiale

Terminaison

- L'étape 1 et l'étape 5 prennent au maximum $i-1$ opérations
- L'étape 2 et l'étape 4 prennent correspondent à 1 opération chacune
- L'étape 3 correspond à 2 opérations

Soit au total $2i+2$ opérations pour un chemin de longueur i bases entre la base du trou et l'une des deux bases considérée.

Echanger deux pions *a* et *b* sur deux bases voisines

On considère les deux bases voisines

- 1er cas : le trou se trouve sur l'une des deux bases. On peut aisément échanger les pions *a* et *b* en 2 opérations.
- 2e cas : le trou ne se trouve pas sur l'une des deux bases.
  1. « déplacer le trou »sur une base voisine des deux bases considérées
  2. déplacer un pion autre que *a* ou *b* afin de se ramener au 1er cas
  3. 1er cas
  4. remettre le pion de l'étape 2 à sa place
  5. remettre les pions déplacés à l'étape 1 à leur place en « ramenant le trou »à sa position initiale

Echanger deux pions *a* et *b* sur deux bases voisines

On considère les deux bases voisines

- 1er cas : le trou se trouve sur l'une des deux bases. On peut aisément échanger les pions *a* et *b* en 2 opérations.
- 2e cas : le trou ne se trouve pas sur l'une des deux bases.
    1. « déplacer le trou »sur une base voisine des deux bases considérées
    2. déplacer un pion autre que *a* ou *b* afin de se ramener au 1er cas
    3. 1er cas
    4. remettre le pion de l'étape 2 à sa place
    5. remettre les pions déplacés à l'étape 1 à leur place en « ramenant le trou »à sa position initiale

A partir de la primitive **Echanger deux pions *a* et *b* sur deux bases voisines**

- on peut se ramener à un algorithme déjà connu **"tri à bulles"**
- proposer une nouvelles primitive **Echanger deux pions *a* et *b* sur deux bases quelconques**

Echanger deux pions a et b sur deux bases quelconques

On suppose que l'on sait échanger deux pions a et b pour un chemin de longueur i bases entre les bases des deux pions a et b
On considère un chemin de longueur $i+1$ bases entre les bases des deux pions a et$b$

1. échanger le pion a avec un pion sur la base voisine
2. échanger les pions a et b pour un chemin de longueur i bases
3. échanger le pion b avec le pion de l'étape 1

Algorithme utilisant la primitive **échanger deux pions *a* et *b* sur deux bases quelconques**

Répéter pour chaque base :

1. Répéter pour chaque pion qui n'est pas de la couleur de la base :
    1.1 échanger le pion avec un pion qui n'est pas sur sa base

Conclusion, terminaison de l'algorithme

Certains algorithmes ne se terminent pas comme. Un tel algorithme ne peut donc pas être considéré comme une réponse satisfaisante au problème posé.

Pour montrer qu'un algorithme ne convient pas on trouve un **contre-exemple**.

Pour montrer qu'un algorithme est correct il est possible de :

- **Tester sur tous les cas** : on vérifie que l'algorithme trouve la solution au problème pour tous les cas possibles.
- Ecrire une **preuve "mathématique"** : on prouve que l'algorithme fonctionne pour le cas général.
- Montrer qu'on peut **se ramener à un algorithme classique** celui-ci ayant été prouvé par ailleurs.

Conclusion

Les activités d'informatique débranchées permettent aux élèves de tous niveaux :

- s'approprier un problème, étudier éventuellement un problème plus simple
- décomposer un problème en sous-problèmes
- dégager des principes généraux, faire des conjectures
- construire des algorithmes
- étudier ces algorithmes avec des exemples ou contre-exemples à leur portée
- établir des preuves partielles ou complètes de ces algorithmes

Références bibliographiques

- Sege Abiteboul, Gilles Dowek. Le temps des algorithmes, édition Le Pommier., 2017
- Sylvie Alayrangues, Samuel Peltier, Laurent signac. Informatique débranchée: construire sa pensée informatique sans ordinateur,Colloque Mathématiquesen Cycle3 IREM de Poitiers, Jun 2017, Poitiers , France. pp.216-226.*
- Christian P. Brackmann, Marcos Román-González, Gregorio Robles, Jesús Moreno-León, Ana Casali, Dante Barone. Development of Computational Thinking Skills through Unplugged Activities in Primary School, Proceedings of the 12th Workshop on Primary and Secondary Computing Education - WiPSCE '17., 2017

Références bibliographiques, suite

- Collectif, Informatique débranchée, Tangente éducation numéro 42-43, 2017
- Gilles Dowek. Les quatre concepts de l'informatique. Sciences et technologies de l'information et de la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques., Oct 2011, Patras,Grèce. pp.21-29.*
- Tim Bell,Ian H. Witten et Mike Fellows,VF équipe interstices. Computer Science Unplugged, 2014*

Sitographie

- Site d'histoire des mathématiques de Bernard Ycart*
- Site de CSUnpllugged*
- IREM Grenoble*
- IREM Clermont-Ferrand*
- ISO Université Grenoble Alpes*
- Page médiation scientifique de Marie Duflot*
- Page médiation scientifique de Martin Quinson*