

Algorithmique 1

Instructions et casse-têtes

Email : Christian.Gresse@univ-orleans.fr

Email : David.Teller@univ-orleans.fr

Au cours de ce TD, vous allez travailler sur vos premiers algorithmes, à l'aide de casse-têtes simples. Pour commencer, nous emploierons des langages d'instructions linéaires, c'est-à-dire dans lesquels l'algorithme n'a jamais à prendre de décisions. Ces langages seront enrichis au cours des semaines qui suivent, jusqu'à parler de Java, l'un des langages de programmation les plus utilisées de nos jours.

Note N'hésitez pas à poser des questions, au professeur ou à son humble TD-man.

Note Lisez rapidement tout l'énoncé avant de répondre aux questions.

1 Formalisations

Pour aujourd'hui, nous emploierons les définitions suivantes :

Définition 1. *Un langage d'instructions (ou jeu d'instructions) est un ensemble de phrases. On appelle chacune de ces phrases une **instruction**.*

Exemple 2. Le langage d'instructions du Sudoku est composé de toutes les phrases « inscrire le nombre n dans la case (x, y) », où n est un nombre entier entre 1 et 9 et x et y sont des nombres entiers entre 1 et 9.

Exemple 3. Le langage d'instructions du Morpion est composé de

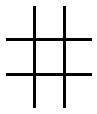
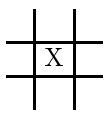
- commencer la partie
- toutes les phrases « inscrire un X dans la case (x, y) », où x et y sont des nombres entiers entre 1 et 3 ;
- toutes les phrases « inscrire un O dans la case (x, y) », où x et y sont des nombres entiers entre 1 et 3 ;
- déclarer la victoire du joueur X
- déclarer la victoire du joueur O .

Définition 4. *Une fois défini un langage d'instructions, un **algorithme** est une suite précise d'instructions.*

Exemple 5. Dans le langage d'instructions du Sudoku, ce qui suit est un algorithme :

1. inscrire le nombre 7 dans la case $(1, 1)$
2. inscrire le nombre 3 dans la case $(3, 8)$
3. inscrire le nombre 9 dans la case $(2, 2)$.

Exemple 6. Dans le langage d'instructions du Morpion, ce qui suit est un algorithme :

1. commencer la partie 
2. inscrire un X dans la case $(2, 2)$ 

3. inscrire un O dans la case $(1, 2)$

O	X	

.

Exemple 7. Dans le langage d'instructions du Morpion, ce qui suit est un algorithme – l'algorithme est faux, car il ne respecte pas les règles du jeu, mais reste un algorithme.

1. inscrire un X dans la case $(2, 2)$
2. inscrire un O dans la case $(2, 2)$.

Note on numérote souvent les étapes d'un algorithme pour pouvoir s'y référer plus facilement. On utilise aussi fréquemment des abréviations pour éviter d'avoir à recopier à chaque fois des instructions complexes.

2 Casse-têtes

2.1 Bruce Willis contre les bidons d'eau

Dans cet exercice, le vaillant Bruce Willis, confronté à une infââme menace terroriste sur le parc municipal de New York, doit trouver le moyen de placer sur une balance un bidon rempli d'exactly 4 litres d'eau. Malheureusement pour lui, il ne dispose que d'une fontaine d'eau et des bidons suivants :

- un bidon de 2 litres (bidon A)
- un bidon de 3 litres (bidon B)
- un bidon de 5 litres (bidon C).

Le langage d'instructions dont il dispose est le suivant :

- « remplir totalement le bidon X » (où X est A , B ou C)
- « vider totalement le bidon X » (où X est A , B ou C)
- « transférer autant que possible du contenu du bidon X dans le bidon Y » (où X est A , B ou C , Y est A , B ou C et X et Y ne sont pas le même bidon)
- « peser le bidon X » (où X est A , B ou C).

L'algorithme sera correct s'il contient une seule pesée, s'il s'achève par une pesée et si le bidon pesé contient exactement 4 litres d'eau. On suppose que les bidons sont initialement tous vides.

Exercice 1. Écrivez un algorithme correct pour permettre à Bruce Willis de sauver Central Park.

Exercice 2. Même chose mais sans utiliser le bidon A .

2.2 Chèvre vs. Chou vs. Loup

Cet exercice se déroule le long d'une rivière. Un berger désire faire traverser à la rivière sa chèvre, son chou et son loup. Oui, il a un loup. Pour ce faire, il dispose en tout et pour tout de ses maigres épaules, à même de transporter à chaque fois soit la chèvre, soit le chou, soit le loup. Bien entendu, il ne peut pas laisser ensemble sans surveillance la chèvre et le chou – sous peine de voir la première engloutir le deuxième – ni la chèvre et le loup – sans quoi ce dernier se fera une joie de déguster sa proie.

En d'autres termes, le jeu d'instruction dont il dispose est le suivant :

- « prendre sur ses épaules X » (où X est Chèvre, Chou ou Loup)
- « déposer X » (où X est Chèvre, Chou ou Loup)
- « aller à la rive de gauche »
- « aller à la rive de droite ».

L'algorithme sera correct si

- la chèvre et le chou ne sont jamais laissés ensemble sur une rive sans berger pour les surveiller
- la chèvre et le loup ne sont jamais laissés ensemble sur une rive sans berger pour les surveiller
- le berger n'a jamais plus d'une chose sur le dos
- à la fin de l'algorithme, tout le monde, berger y compris, est sur la rive de droite.

On supposera que tout le monde commence sur la rive de gauche.

Exercice 3. Écrivez un algorithme correct pour résoudre le problème.

2.3 La charge de la cavalerie légère

Nous voici maintenant sur un plateau d'échecs, sur lequel se déplace, seul, un cavalier.

Le cavalier peut :

- « avancer de deux cases vers les lignes adverse puis d'une case vers la droite »
- « avancer de deux cases vers les lignes adverse puis d'une case vers la gauche »
- « avancer de deux cases vers les lignes amies puis d'une case vers la droite »
- « avancer de deux cases vers les lignes amies puis d'une case vers la gauche »
- « avancer de deux cases vers la gauche puis d'une case vers les lignes amies »
- « avancer de deux cases vers la gauche puis d'une case vers les lignes adverses »
- « avancer de deux cases vers la droite puis d'une case vers les lignes amies »
- « avancer de deux cases vers la droite puis d'une case vers les lignes adverses ».

Nous allons essayer de faire faire quelques figures à ce cavalier. Nous supposons que le plateau de jeu est vide, à l'exception du cavalier. Un algorithme est correct si le cavalier ne sort jamais du plateau.

Exercice 4. Écrivez un algorithme pour déplacer le cavalier (probablement en plusieurs sauts) d'une case vers les lignes adverses.

Exercice 5. Écrivez un algorithme pour déplacer le cavalier (probablement en plusieurs sauts) d'une case vers les lignes adverses et d'une case vers la droite.

Exercice 6. Dans quelles circonstances les algorithmes précédents sont-ils corrects ?