

# Examen d'algorithmique

## Session principale 2016-17

Date : 16-01-2017

- 
- **Durée** : 2 heures.
  - Les exercices sont indépendants.
  - Aucun document n'est autorisé.
  - Barème indicatif (Total : 22 points) : Exercice 1 : 5 points, Exercice 2 : 5 points, Problème : 12 points.
- 

### 1 Exercice 1 : Analyse d'un algorithme

Considérons l'algorithme suivant :

---

— Données : un réel  $A$ , un entier  $N > 0$ .

---

- $a := A$
  - $n := N$
  - $r := 1$
  - **Tant que** ( $n > 0$ )
  - **Début**
    - **Si** ( $n \bmod 2 = 0$ ) **Alors**
      - $a := a * a$
      - $n := n \text{ div } 2$
    - **Sinon**
      - $r := r * a$ ;
      - $n := n - 1$ ;
  - **Fin**
- 

1. Exécutez cet algorithme sur un exemple de votre choix.
2. Quelle est la valeur  $r$  calculée par cet algorithme.

3. Démontrez à l'aide d'un invariant de boucle que cet algorithme calcule bien cette valeur.

## 2 Exercice 2 : Points, systèmes et centres de gravité

Un **point** est représenté par la structure suivante :

— **Enregistrement** *Point*

—  $x$  : Réel

—  $y$  : Réel

—  $z$  : Réel

— **Finenregistrement**

Un **système** est un ensemble de points ayant chacun un poids. Il est défini par une Map. Les clefs de cette Map sont les points, ses valeurs sont les poids.

1. Écrire un algorithme permettant de calculer le centre de gravité d'un système.
2. Écrire un algorithme permettant de calculer le centre de gravité de  $N$  systèmes stockés dans un tableau *tableauSystemes*. Chaque système aura un poids associé qui sera stocké dans un tableau *tableauPoidsSystemes*. Les deux tableaux ont le même nombre d'élément. L'index d'un système et l'index du poids qui lui est associé seront identiques dans les deux tableaux.

**Rappel : définition du centre de gravité**

Pour  $n$  points  $A_i$ ,  $1 \leq i \leq n$  le centre de gravité  $G$  est calculé par :

$$x_G = \frac{a_1x_1 + \dots + a_nx_n}{a_1 + \dots + a_n}, \quad y_G = \frac{a_1y_1 + \dots + a_ny_n}{a_1 + \dots + a_n}, \quad z_G = \frac{a_1z_1 + \dots + a_nz_n}{a_1 + \dots + a_n}.$$

## 3 Problème : le sous-tableau maximal (d'après [1])

Dans cet exercice, nous considérons le problème suivant :

- Étant donné un tableau *tab* de taille  $n + 1$ , et dont les éléments sont des entiers positifs, trouver les indices  $i_1$  et  $i_2$  tels que :
  - $1 \leq i_1 < n + 1$
  - $i_1 < i_2 \leq n + 1$
  - $tab[i_2] - tab[i_1]$  = maximum des écarts possibles pour toutes les valeurs de  $i_1$  et  $i_2$ .

- **Application :** On peut penser aux valeurs  $tab[i]$  comme étant la valeur en bourse d'une action. Dans ce cas,  $tab[i_1]$  et  $tab[i_2]$  représentent respectivement le prix auquel on achète l'action et celui auquel on la vend. L'objectif étant de maximiser le profit (prix de vente - prix d'achat).
- 1. Écrire un algorithme 'naïf' pour résoudre ce problème. On qualifie de 'naïf' un algorithme qui envisage toutes les possibilités pour  $i_1$  et  $i_2$ .
- 2. Quelle est la complexité de cet algorithme ?

Considérons à présent le tableau  $tab2$  défini à partir de  $tab$  de la manière suivante :

- la taille de  $tab2$  est de  $n$ .
- $tab2[i] = tab[i + 1] - tab[i]$ .

3. Donnez un exemple de tableau  $tab$  et donnez le tableau  $tab2$  associé.
4. Vérifiez que le problème que nous cherchons à résoudre est équivalent au problème suivant "Trouver dans le tableau  $tab2$  un sous-tableau connexe (composé de cases successives) dont la somme des éléments est maximale" .

C'est ce dernier problème que nous appelons le **problème du sous tableau maximal**. Notre objectif est à présent de trouver un algorithme utilisant l'approche **diviser pour régner** pour cette dernière version du problème.

5. Soient donc  $bas$  et  $haut$  deux indices du tableau tels que  $bas \leq haut$  et  $mil$  le milieu de ces deux indices. Appelons  $P(bas, haut)$  le problème qui consiste à trouver le sous-tableau maximal sur  $tab2[bas..haut]$  (le problème initial est donc  $P(1, n)$ ). Comment peut-on exprimer le problème  $P(bas, haut)$  en utilisant l'approche diviser pour régner ?
6. Écrire une procédure *Trouver-Sous-Tableau-Milieu* permettant de résoudre le problème auxiliaire suivant :
  - $Q(bas, haut)$  : trouver le sous-tableau maximal parmi les sous-tableaux de  $tab[bas..haut]$  qui ne sont pas entièrement inclus dans  $tab[bas..mil]$  ou  $tab[mil + 1..haut]$ . Autrement dit, le problème  $Q(bas, haut)$  considère exclusivement les sous-tableaux  $tab[p..q]$  tels que  $p \leq mil < mil + 1 \leq q$ .
7. Calculez la complexité de cette procédure.
8. Écrire l'algorithme récursif pour la résolution du problème du sous-tableau maximal.
9. Donnez la formule permettant de calculer la complexité de cet algorithme.

10. Déduisez-en cette complexité.

## Références

- [1] Cormen T.H., Leiserson C.E., Rivest R.L. et Stein C. Algorithmique. Dunod. 2010, 1188 pages.