

## Examen d'algorithmique

Vendredi 6 janvier 2012 8h30/11h30 – Aucun document autorisé

Mode d'emploi : Le barème est donné à titre indicatif.

La qualité de la rédaction des algorithmes et des preuves sera très fortement prise en compte.

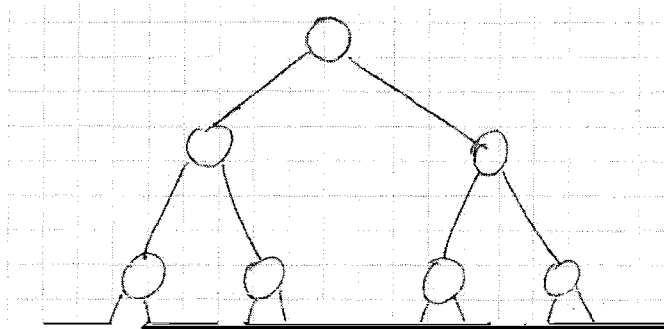
1. Montrer que l'on a  $n_{\min} = n_f - 1$ . Quelle est la condition sur  $n_f$  pour que les feuilles se situent sur un seul niveau de l'arbre ?

Montrer que la hauteur d'un arbre Min est inférieure ou égale à  $\lceil \log(n_f) \rceil$  (on suppose que  $n_f \geq 1$ ).

Pour représenter un arbre Min, on utilise un tableau d'entiers  $T[-]$  de taille  $2n_f - 1$  et commençant à l'indice 1 : la valeur associée au noeud  $i$  est stockée en  $T[i]$ . Cela permet d'obtenir facilement le père ( $i/2$ ), le fils gauche ( $2i$ ) et le fils droit ( $2i + 1$ ) du noeud  $i$  (NB : on utilise la même idée que pour les tas). L'entier stocké sur un noeud sera soit sa valeur initiale si c'est une feuille, soit le min des valeurs des fils.

2. Donner un algorithme (efficace !) `CalculValArbre` qui, étant donné un arbre Min  $T[1 \dots 2n_f - 1]$  où seules les feuilles (c.-à-d. les noeuds  $n_f, n_f + 1, \dots, 2n_f - 1$ ) ont été initialisées à leur valeur, calcule la valeur de tous les noeuds "min". Donner sa complexité.

Appliquer votre algorithme à l'exemple de la figure 2.



Appliquer votre algorithme à l'exemple de la figure 2.

5. Etant donné un arbre Min obtenu avec `CalculArbre( $T$ )`, trouver le deuxième plus petit élément en utilisant `MaJ`.
6. Etendre l'idée précédente pour obtenir un algorithme qui trouve le  $k$ -ème plus petit élément (avec  $k \in \{1, \dots, n\}$ ). Donner sa complexité.
7. On considère à présent l'algorithme suivant où  $V$  est un tableau d'entiers de taille  $n$  et  $k$  est un entier vérifiant :  $1 < k \leq n$ .

**Algo**( $V[1 \dots n], k$ ) :

- 1 : Appliquer `CalculArbre` sur un arbre avec les  $n - k + 2$  feuilles  $V[1], \dots, V[n - k + 2]$ .
- 2 : Pour  $i = 1$  à  $k - 2$  faire :
  - 3 : Appliquer `MaJ`( $T, V[n - k + 2 + i], T[1]$ )
- 4 : Renvoyer le deuxième plus petit élément de l'arbre  $T$ .

**Remarque :** On suppose que l'instruction 4 est basée sur l'algorithme de la question 5.

- (a) Appliquer `Algo` au tableau  $V = [12, 3, 5, 6, 7, 22, 35, 8, 18, 2, 56, 89, 4, 9]$  et  $k = 5$ .
  - (b) Que fait cet algorithme ? Expliquer le (montrer qu'il est correct).
  - (c) Quelle est sa complexité ?
8. Comparer les algorithmes des questions 6 et 7.
  9. Comparer l'algorithme `Algo` de la question 7 avec les deux autres vus en cours (le premier était une simple variante de quicksort, et le second utilisait comme pivot le médian des médians des "paquets de 5 éléments" ...).

### Exercice 1 : Remplir un camion (5 points)

On dispose de  $n$  ( $n \geq 1$ ) objets ayant des poids  $p_1, p_2, \dots, p_n$  et des valeurs  $v_1, v_2, \dots, v_n$ . On suppose que les  $p_i$  et  $v_i$  sont des entiers.

Dans les questions ci-dessous, on cherche des algorithmes (efficaces !) pour charger un camion dont la capacité maximale est donnée par un poids  $K$  en cherchant à maximiser la valeur des objets transportés. Pour chaque question, on demande d'expliquer l'algorithme proposé et de donner sa complexité.

(Citer les algorithmes)

1. On suppose que  $v_i = 1$  pour tout  $i = 1, \dots, n$ .
2. On suppose que  $v_i = p_i$  pour tout  $i = 1, \dots, n$ .
3. Cas général : sans contrainte sur les  $v_i$ .

### Exercice 2 : Flots (5 points)

1. On considère le réseau de transport  $R$  de la figure 3. Appliquer l'algorithme de Ford-Fulkerson pour trouver un flot maximal. On donnera tous les flots intermédiaires et les graphes des augmentations obtenus par l'algorithme.

En déduire une coupe de capacité minimale du réseau  $R$ .

2. Appliquer l'algorithme de Ford-Fulkerson sur le même réseau  $R$  que la question précédente mais en choisissant d'autres chemins dans les graphes des augmentations de manière à obtenir le résultat avec un nombre d'itérations différent.

