ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 6 1 November 2010

1. Running time

Soit $f(n)$ la fonction donnée ci-dessous en pseudo code:

```
Call: f(n)
1: a ← 0
2: b ← ln(n)
3: for i = 1, ..., n do
4:     a ← a + b
5: for j = 1, ..., n do
6:     for k = 1, ..., j do
7:         for ℓ = j + 1, ..., j + n do
8:             a ← a + b
9: return a
```

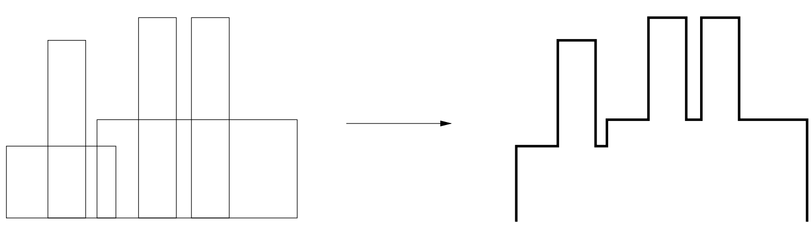
1. Trouver une formule fermée pour $f(n)$.

2. Trouver s et t tels que
$$f(n) = \theta\big(n^s \cdot \ln(n)^t\big).$$

3. Quel est, en notation θ, le temps de parcours de cet algorithme sachant que la ligne 2 se fait en temps $\theta(1)$?

2. Diviser pour régner: Le problème du skyline

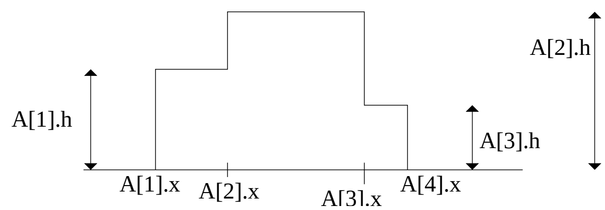
On décrit un bâtiment par un triplet $(x_1, x_2, h) \in \mathbb{R}^3$ où x_1 décrit la première coordonnée sur l'axe x et x_2 la deuxième. La hauteur du bâtiment est décrite par h. Etant donné une liste de n bâtiments, la tâche est de trouver la silhouette (skyline) formée par nos bâtiments. Illustration:



La silhouette est représentée par la liste croissante de coordonnées x et la hauteur correspondante.

a) Etant donné deux skylines on veut écrire un programme pour les combiner (c'est-à-dire trouver le skyline obtenu avec tous les bâtiments du premier et du deuxième).

On suppose qu'on nous donne deux tableaux A et B dont chaque entrée contient deux valeurs: x et h. Ces tableaux décrivent les deux skylines qu'on veut combiner. Le output est aussi un skyline, c'est-à-dire un tableau C avec le même type d'entrées, qui décrit le skyline obtenu en combinant les skylines de A et B.



Ecrire un algorithme qui, étant donné comme input les tableaux A et B, retourne comme output le tableau C. (*Indication:* Regarder MergeSort).

b) Utiliser le fait qu'on puisse combiner deux skylines pour construire un algorithme diviser-pour-régner qui résout le problème Skyline en $O(n \log(n))$.

3. Sac à dos 0/1

a) Trouver une solution optimale (i.e. objets et valeur optimale) au problème du sac à dos 0/1 suivant:

$W = 20$,

objet	A	B	C	D	E
valeur	22	15	8	15	15
poids	11	7	6	7	3

b) La solution optimale dans le problème ci-dessus est-elle unique? Si ce n'est pas le cas, donner toutes les solutions optimales. Peut-on adapter l'algorithme du point a) pour trouver toutes les solutions optimales? Comment faire?

c) Montrer que si on a un problème de sac à dos avec n objets, alors il peut y avoir au plus 2^n solutions optimales.

d) Montrer que pour tout n pair, il y a un choix de W, des v_i et des w_i tel qu'il y a au moins $2^{n/2}$ solutions optimales.