

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
 Sections d'Informatique et de Systèmes de Communication

Série d'exercices 5

24 Oct. 2011

1. Algorithmes récursifs : Le diamètre d'un arbre binaire

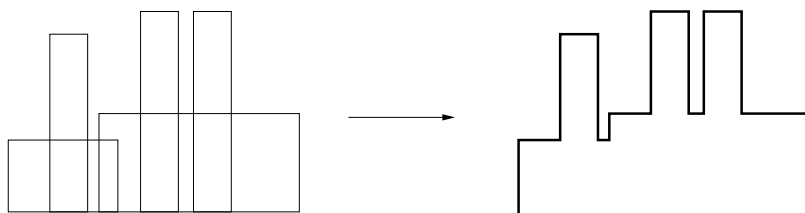
Soit $G = (V, E)$ un graphe. Pour deux sommets $a, b \in V$, on définit la *distance* $d(a, b)$ comme étant la longueur du plus court chemin de a à b . Le *diamètre* de G est défini comme étant la distance maximale dans le graphe, i.e.

$$\text{diam}(G) := \max_{a,b \in V} d(a, b).$$

- a) Soit T un arbre binaire, et soient T_1 et T_2 ses sous-arbres gauche et droit. Exprimer le diamètre puis la hauteur de T en fonction des diamètres et hauteurs de T_1 et T_2 .
- b) Construire un algorithme qui calcule le diamètre d'un *arbre binaire* en temps linéaire (en le nombre de sommets).
 (Indice : Il faut construire par récurrence un algorithme qui, étant donné un arbre binaire, retourne sa hauteur et son diamètre en temps linéaire).

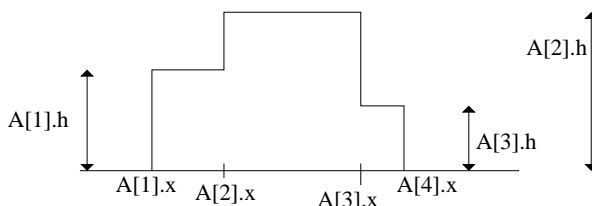
2. Diviser pour régner : Le problème du skyline

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 $O(n \log(n))$.

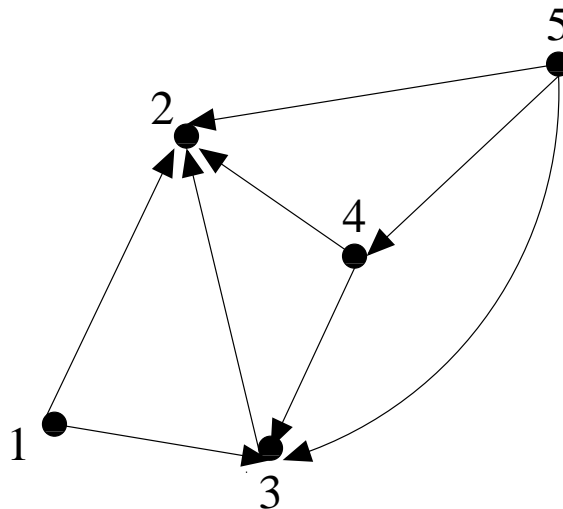
3. Algorithmes par induction : Le problème de la célébrité

Nous aimerions trouver une célébrité dans un ensemble V de n personnes. Pour modéliser ce problème, nous caractérisons une célébrité comme suit : C'est une personne qui est connue de tout le monde, mais qui ne connaît personne.

Pour résoudre ce problème, nous avons le droit de poser un seul type de question : si $a, b \in V$ sont des personnes, nous pouvons demander à a s'il connaît b , et il nous répondra par "oui" ou "non". Le problème est donc le suivant : Etant donné un groupe de personnes V contenant une célébrité, trouver la célébrité en posant de telles questions.

Remarquons qu'on peut modéliser ce problème comme un problème de graphe orienté (V, E) : L'ensemble V est l'ensemble des personnes, et on met une flèche de $a \in V$ à $b \in V$ si et seulement si a connaît b . Poser la question "a connaît-il b?" revient alors à vérifier si l'élément (a, b) de la matrice d'adjacence vaut 1.

- a) Dans le graphe suivant y-a-t il une célébrité ?



- b) Supposons que nous avons un ensemble de personnes V dont certaines se connaissent (on obtient donc un graphe orienté (V, E)). Montrer qu'il peut y a voir au plus une célébrité dans V .
- c) Supposons que le sommet i est une célébrité. Que peut-on dire sur la ligne i et la colonne i de la matrice d'adjacence du graphe ?
- d) Décrire l'algorithme naïf pour résoudre le problème de la célébrité, étant donné la matrice d'adjacence du graphe. Donner l'ordre du nombre de questions qu'il faut poser (c'est-à-dire le nombre d'entrées dans la matrice qu'il faut regarder) en fonction du nombre de personnes $n = |V|$.
- e) Observons que pour tout $a, b \in V$, on a
- Si a connaît b , alors a ne peut pas être une célébrité.
 - Si a ne connaît pas b , alors b ne peut pas être une célébrité.

Nous voulons utiliser ce fait pour construire par induction un algorithme $O(n)$ pour résoudre le problème de la célébrité.

Pour simplifier, nous supposons qu'il y a en effet une célébrité dans G , et qu'il s'agit donc juste de la trouver.

- (i) Base. Montrer que pour $n = 2$ on peut résoudre ce problème en posant une seule question.
- (ii) Pas. Supposons que nous savons résoudre le problème pour n personnes en posant $T(n)$ questions. Montrer qu'on peut le résoudre pour $n + 1$ personnes en posant $T(n) + 1$ questions.
- (iii) En déduire un algorithme pour résoudre le problème pour n personnes, et donner le nombre de questions qui sont nécessaires.