L3 ENS, année 2008-09
Algorithmique et Programmation
Contrôle du 8 décembre 2008
Notes de cours autorisées, à l'exclusion de tout autre document

Exercice 1 : Location de skis

On souhaite spécifier un algorithme efficace pour une attribution optimale de m paires de skis de longueur $s_1, \ldots, s_m$, respectivement, à n skieurs ($m \geq n$) de taille $h_1, \ldots, h_n$, respectivement, via une fonction (injective) $f : \{1, \ldots, n\} \to \{1, \ldots, m\}$, f étant optimale lorsqu'elle minimise $\sum_{k=1}^{n} |s_{f(k)} - h_k|$. Soit $A[n, m]$ ce minimum et $A[i, j]$ pour $i \leq n$ et $j \leq m$ où i et j font référence aux i premiers skieurs et aux j premières paires de skis, respectivement.

1. Supposons que les hauteurs des skis et des skieurs soient triées par ordre croissant. Montrer la propriété suivante :

   (P) *Si* $f : \{1, \ldots, i\} \to \{1, \ldots, j\}$ *(injective) est telle qu'il existe* $i_1 \leq i$ *avec* $f(i_1) = j$, *alors* $\sum_{k=1}^{i} |s_{f(k)} - h_k| \geq \sum_{k=1}^{i} |s_{f'(k)} - h_k|$, *avec* f' *obtenue à partir de* f *en échangeant les images de* i_1 et i *(donc* $f'(i_1) = f(i), f'(i) = j$ *et* $f' = f$ *ailleurs).*
2. Définir $A[i, j]$ en fonction de valeurs plus petites $i \leq n$ et $j \leq m$ (lesquelles ?), par une équation de récurrence.
3. Analyser la complexité en veillant à affiner l'analyse de sorte à garantir que l'algorithme soit en $O(n \log n)$ si $m = n$.
4. Montrer qu'on peut avoir une meilleure complexité lorsque $n^2 = o(m)$. (Indication : se restreindre à $O(n^2)$ paires de skis.)

Arbre Couvrant Minimal en temps linéaire probabiliste

On se propose d'étudier un algorithme probabiliste qui permet de calculer l'arbre couvrant de poids minimal en temps linéaire. Dans tout le problème, on suppose que $G = (X, E, w)$ est un graphe non-orienté connexe ayant n sommets et m arêtes et où chaque arête e a un poids $w(e)$. Pour simplifier, on supposera que tous les poids sont différents, il existe ainsi un seul arbre couvrant de poids minimal.

Calcul par contraction d'arêtes

On rappelle que la contraction de l'arête $E = \{x, y\}$ consiste à remplacer les deux sommets x et y par un seul sommet z et les arêtes incidentes à x et y par des arêtes incidentes à z. Noter que les arêtes multiples peuvent être créées par une contraction.

Pour chaque sommet x de G, on note $e_{min}(x)$ l'arête de poids minimal qui est incidente à x et V_{min} l'ensemble de toutes les $e_{min}(x)$.

1. Montrer que pour tout cycle C d'un graphe, l'arête de poids la plus lourde de C n'apparaît pas dans l'arbre couvrant minimal.
2. Montrer que toute arête de V_{min} appartient à l'arbre couvrant de poids minimal de G. (Indication : raisonner à partir de l'algorithme glouton qui construit l'arbre en prenant les arêtes de V_{min} par ordre croissant.)
3. Déterminer les valeurs α_n et β_n telles que

$$\alpha_n \leq |V_{min}| \leq \beta_n.$$

   Proposer deux exemples de poids sur le cycle à 6 sommets pour lesquels ces bornes sont atteintes.

   Soit $Contr(G)$ le graphe obtenu à partir de G en contractant *toutes* les arêtes de V_{min}.
4. Concevoir un algorithme en temps $O(m + n)$ qui calcule $Contr(G)$. (Indication : déterminer les composantes connexes des arêtes de V_{min} et les remplacer par un seul sommet. Penser à éliminer les boucles et les arêtes multiples créées par la contraction.)
5. Montrer que la donnée de l'arbre couvrant de poids minimal pour $Contr(G)$ permet de construire l'arbre de poids minimal pour G.
6. En déduire un algorithme récursif qui construit l'arbre de poids minimal d'un graphe. Quelle est sa complexité ?

Échantillonage

À partir d'un graphe G, on construit un graphe $G_{/2}$ en effectuant un tirage à pile ou face équilibré sur chaque arête de G : une arête de G est dans $G_{/2}$ avec probabilité 1/2.

1. Quel est le nombre moyen d'arêtes dans $G_{/2}$?

   Le graphe $G_{/2}$ n'étant pas nécessairement connexe, on considère sa *forêt couvrante* F de poids minimal ; elle est constituée des arbres couvrants de poids minimal de chacune de ses composantes connexes. Une arête $e = \{x, y\}$

de G est dite *lourde* pour $G_{/2}$ s'il existe dans F un chemin de x à y composé d'arêtes toutes de poids inférieur à $w(e)$. Une arête qui n'est pas lourde est dite *légère*.

2. Que peut-on dire d'une arête lourde par rapport à l'arbre couvrant de poids minimal de G ? et d'une arête légère ?

   On considère que le processus suivant qui considère les arêtes e de G par ordre de poids croissant : à chaque pas un bit aléatoire est tiré ; l'arête e est ajoutée dans $G_{/2}$ si et seulement si le bit est égal à 1 (probabilité 1/2).

3. Donner un algorithme qui construit itérativement au cours du tirage les ensembles d'arêtes suivants : F, la forêt couvrante minimale de $G_{/2}$ et L, l'ensemble des arêtes légères de G qui ne sont pas dans $G_{/2}$.

   On peut considérer le processus aléatoire ci-dessus comme composé de k phases, chacune constituée des tirages entre deux ajouts d'un éléments dans F. Au cours de chaque phase sont tirées des arêtes lourdes et des arêtes légères.

4. Montrer que le nombre d'éléments de $F \cup L$ est majoré en moyenne par $2n$.

Algorithme

On utilise l'algorithme suivant de détermination d'un arbre couvrant de poids minimal :

`MSTProba(G)`

1. `Appliquer 3 fois à la suite l'algorithme de contraction` V_{min} `; on obtient un graphe` G_1 `et des ensembles d'arêtes` V_1, V_2, `et` V_3.
2. `Construire` $G_2 = G_{1/2}$.
3. `Soit` $F_1 =$ `MSTProba`(G_2).
4. `Soit` G_3 `obtenu à partir de` G_2 `en ne conservant que` F_1 `et les arêtes légères de` G_1.
5. $T_2 =$ `MSTProba`(G_3).
6. `return` $T_2 \cup V_1 \cup V_2 \cup V_3$.

1. Exprimer un majorant du nombre de sommets de G_2, G_3 et du nombre moyen d'arêtes de G_3 en fonction de n. Majorer le nombre moyen d'arêtes de G_2 en fonction de m.
2. On admet que le nombre d'opérations de l'étape 4 de l'algorithme est linéaire en $(n + m)$. En déduire une formule de majoration du nombre d'opérations moyen $T(m, n)$ de `MSTProba(G)` qui fait intervenir des $T(m', n')$ et une fonction linéaire $c(m + n)$.
3. Montrer par récurrence que $T(m, n) \leq 2c(m + n)$.