TD 12 – Préparation à l'examen

Ces exercices sont issus du sujet d'examen donné en 2017 pour le cours Algorithmique 1 à l'ENS de Lyon.

Exercice 1. *Amusettes Algorithmiques*

Partition en paires. Soient n entiers $a_1, \cdots, a_n$ avec n pair. On cherche à les partitionner en $n/2$ paires telles que le maximum des sommes des deux éléments de chaque paire soit minimal. Donner la solution optimale avec une petite preuve de correction.

Points maximaux. Soit S un ensemble de n points entiers du plan (x_i, y_i), pour $1 \leq i \leq n$, à coordonnées distinctes deux à deux. On dit qu'un point (x, y) domine un point (x', y') si $x > x'$ et $y > y'$. Un point de S est dit maximal s'il n'est dominé par aucun autre point de S. Proposer un algorithme efficace pour déterminer les points maximaux de S.

Stockage. Soient n programmes P_i, pour $1 \leq i \leq n$, à stocker sur un disque de capacité totale D. Le programme P_i occupe une place d_i, avec $D < \sum_{i=1}^{n} d_i$. Peut-on utiliser un algorithme glouton dans chacune des situations suivantes :

1. on veut maximiser le nombre de programmes stockés sur le disque ;
2. on veut minimiser l'espace disque inutilisé.

Exercice 2. *Amusettes de NP-complétude*

Montrer que les problèmes suivants sont NP-complets :

Roue. Étant donnés un graphe $G = (V, E)$ et un entier $K \geq 3$, déterminer si G contient une roue de taille K, i.e. un ensemble de $K + 1$ sommets w, $v_1, v_2, \cdots, v_K$ tels que $(v_i, v_{i+1}) \in E$ pour $1 \leq i < K$, $(v_K, v_1) \in E$ et $(v_i, w) \in E$ pour $1 \leq i \leq K$ (w est le centre de la roue).

2-Partition-Approx. Étant donnés n entiers strictement positifs $a_1, a_2, \cdots, a_n$, trouver $I \subset [1, n]$ un sous-ensemble tel que $|\sum_{i \in I} a_i - \sum_{i \notin I} a_i| \leq 1$.

Vertex-Cover-Impair. Étant donné un graphe $G = (V, E)$ dont tous les sommets sont de degré impair, et un entier K, existe-t-il un sous-ensemble W de V de cardinal au plus K et tel que toutes les arêtes de E aient (au moins) une extrémité dans W ? Même question pour Vertex-Cover-Pair, la variante où tous les sommets sont de degré pair.

Exercice 3. *Cardinal minimal d'une vertex cover pour un arbre*

Soit $G = (V, E)$ un graphe. On veut calculer le cardinal minimal d'un sous-ensemble W de sommets couvrant toutes les arêtes (tel que toutes les arêtes aient (au moins) une extrémité dans W).

1. Donner un algorithme de programmation dynamique pour résoudre le problème si G est un arbre. Quelle est sa complexité ?
2. Comment retrouver un sous-ensemble W de cardinal minimal ?

Exercice 4. *Placement de serveurs dans les arbres*

On s'intéresse au placement de serveurs dans les arbres. Un arbre a N nœuds internes $n_1, \cdots, n_N$ (dont la racine) et F feuilles $f_1, \cdots, f_F$. Chaque feuille f_i, pour $1 \leq i \leq F$ porte un nombre entier r_i de requêtes. On peut placer un serveur de capacité W sur n'importe quel nœud interne (y compris la racine). Chaque feuille f devra être associée à un server serveur(f) qui doit être un nœud interne sur le chemin (unique) qui mène de la feuille à la racine. On va chercher à minimiser le nombre de serveurs avec la contrainte de capacité suivante : la somme des requêtes servies par un serveur donné ne doit pas dépasser W : si on a placé un serveur sur le nœud n_j, alors

$$\sum_{\text{serveur}(f_i)=n_j, 1\leq i\leq F} r_i \leq W$$

Pour choisir le serveur de chaque feuille, on considère deux règles possibles :

CLOSEST : dans cette stratégie, le serveur de chaque feuille doit être le premier serveur rencontré quand on remonte le chemin (unique) de la feuille vers la racine ;

UP : dans cette stratégie moins contraignante, le serveur de chaque feuille peut être n'importe quel serveur rencontré quand on remonte le chemin (unique) de la feuille vers la racine.

Dans chaque cas, on cherche à placer un nombre minimal de serveurs, tout en respectant la contrainte de capacité et la règle de choix de la stratégie. Pour un arbre donné $\mathcal{A}$, on note $CLOSEST(\mathcal{A})$ le nombre minimal de serverus à placer avec la stratégie CLOSEST. On définit de même $UP(\mathcal{A})$ pour la stratégie UP.

1. On considère l'arbre $\mathcal{A}$ suivant, à 3 sommets et 3 feuilles, de racine n_1, et on pose $W = 10$:

$$\begin{array}{ccccc} & & & & n_3 - f_3 \\ & & & \diagup & \\ & & n_2 - f_2 & & \\ & \diagup & & & \\ n_1 - f_1 & & & & \end{array}$$

   — Montrer que $CLOSEST(\mathcal{A}) = 3$ et $UP(\mathcal{A}) = 2$ si $r_1 = 3$, $r_2 = 9$ et $r_3 = 7$.
   — Déterminer $CLOSEST(\mathcal{A})$ et $UP(\mathcal{A})$ si $r_1 = 3$, $r_2 = 9$, et $r_3 = 8$.

2. Montrer que le problème suivant est NP-complet : étant donnés un arbre $\mathcal{A}$ et un entier K, est-ce que $UP(\mathcal{A}) \leq K$?

3. **(difficile)** Donner un algorithme polynomial (en la taille $N + F$ de $\mathcal{A}$) calculant $CLOSEST(\mathcal{A})$.

4. **(difficile)** Pour tout K arbitrairement grand, montrer qu'il existe un arbre $\mathcal{A}$ tel que $\frac{CLOSEST(\mathcal{A})}{UP(\mathcal{A}} \geq K$.

Exercice 5. *Dettes entre amis (difficile)*

Des amis $A_1, A_2, \cdots, A_n$ sont partis en vacances, et c'est l'heure des comptes. On connait la matrice M des dettes : $M(i, j)$ est la somme que doit A_i à A_j, positive si A_i doit des sous à A_j et négative dans le cas contraire. Bien sûr on n'a besoin que de la partie triangulaire supérieure de M, à savoir les $M(i, j)$ avec $i < j$ (puisque $M(i, i) = 0$ et $M(j, i) = -M(i, j)$). On cherche à *apurer* [1] les dettes en réalisant des transactions. Une transaction est un montant d'argent non nul donné par un ami A_i à un autre A_j.

Petit exemple. Voici un exemple avec $n = 4$: $M(1, 2) = 10$, $M(1, 3) = 20$, $M(2, 3) = M(3, 4) = -15$ et tous les autres $M(i, j)$ avec $i < j$ sont nuls. Dessinez un petit graphe à 4 sommets pour vous aider. Quel est le nombre minimal de transactions ? Quel est le volume minimal d'argent qui doit être échangé ?

Transactions. On cherche à apurer les dettes en minimisant le nombre de transactions. Montrer que le problème de décision associé est NP-complet.

Volume. On cherche maintenant à apurer les dettes en transférant le volume minimal d'argent, donc en minimisant la somme des montants des transactions, indépendamment du nombre de transactions. Proposer un algorithme glouton pour résoudre ce problème, et justifier sa correction.

1. dans un langage plus commun, on dirait *éponger* ; le terme *apurer* implique une notion de précision associée à l'intervention d'un notaire.