Algorithmique I
Année 2008-2009

A. Benoit, B. Depardon
C. Mouilleron, C. Rezvoy

TD n°14 Examen 2008

1 Questions de cours pour la chauffe

Soient P_1 et P_2 deux problèmes de décision, et supposons qu'on connaisse une transformation polynomiale (une réduction) de P_1 en P_2. Répondre aux sept questions suivantes avec un maximum de deux lignes de justification par question.

1. Si $P_1 \in \mathcal{P}$, a-t-on $P_2 \in \mathcal{P}$?
2. Si $P_2 \in \mathcal{P}$, a-t-on $P_1 \in \mathcal{P}$?
3. Si P_1 est NP-complet, P_2 est-il NP-complet ?
4. Si P_2 est NP-complet, P_1 est-il NP-complet ?
5. Si on connaît une transformation polynomiale de P_2 en P_1, P_1 et P_2 sont-ils NP-complets ?
6. Si P_1 et P_2 sont NP-complets, existe-t-il une transformation polynomiale de P_2 en P_1 ?
7. Si $P_1 \in \mathcal{NP}$, P_2 est-il NP-complet ?

2 Quelques réductions

Montrer que les problèmes suivants sont NP-complets :

Question 2.1 Étant donnés $2n$ entiers $a_1, a_2, \ldots, a_{2n}$, trouver un sous-ensemble $I \subset [1..2n]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$.

Question 2.2 Étant donnés $2n$ entiers $a_1, a_2, \ldots, a_{2n}$, trouver un sous-ensemble $I \subset [1..2n]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$ et $|I| = n$.

Question 2.3 Étant donnés $2n$ entiers $a_1, a_2, \ldots, a_{2n}$, trouver un sous-ensemble $I \subset [1..2n]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$, et pour $j = 1..n$, soit on a $a_{2j-1} \in I$ et $a_{2j} \notin I$, soit le contraire.

Question 2.4 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, ..., 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).

Question 2.5 Étant donnés un graphe $G = (V, E)$ et un entier $K \geq 3$, déterminer si G contient un dominateur de cardinal K, i.e. un sous-ensemble $D \subset V$ de cardinal K tel que pour tout sommet $u \in V \setminus D$, il existe $u \in D$ avec $(u, v) \in E$

Question 2.6 *(difficile)* Etant données n baguettes rigides de longueur entières $a_1, a_2, \ldots, a_n$, pouvant être reliées dans cet ordre bout-à-bout par des charnières, et étant donné un entier k, assembler toutes les baguettes de manière qu'en repliant la chaîne obtenue la longueur totale ne dépasse pas k (voir la Figure 1).
Indication : réduire à partir de 2-Partition.

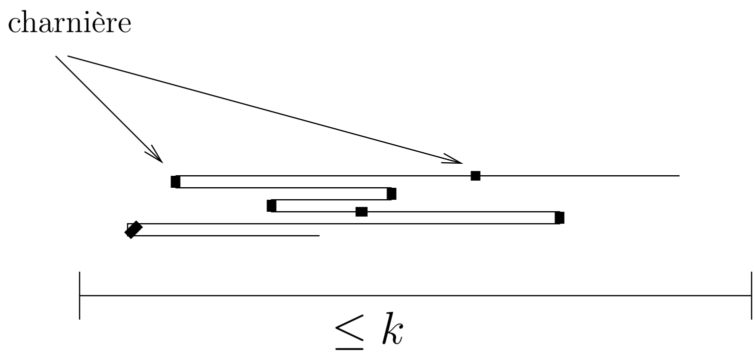

FIG. 1: Pliage des baguettes

3 Recherche d'éléments

On veut une structure de données qui permette la recherche (d'un élément), et l'insertion (d'un nouvel élément) de manière efficace.

Question 3.1 Quels sont les coûts de la recherche et de l'insertion si on utilise un tableau trié de taille n ?

On propose la solution suivante : soit $k = \lceil \log(n+1) \rceil$ et soit $(n_{k-1}, n_{k-2}, \ldots, n_0)$ la représentation binaire de n. On a k tableaux triés $A_0, A_1, \ldots, A_{k-1}$, et la taille de A_i est 2^i pour tout i. Le tableau A_i est plein si $n_i = 1$ et vide si $n_i = 0$. Ainsi le nombre total d'éléments stockés dans les k tableaux est $\sum_{i=0}^{k-1} n_i 2^i = n$. Noter que chaque tableau est trié mais qu'il n'y a aucune relation entre les éléments de deux tableaux.

Question 3.2 Expliquer comment faire une recherche dans cette structure, et donner le coût au pire cas.

Question 3.3 Expliquer comment faire une insertion dans cette structure, et donner le coût au pire cas et en analyse amortie.

Question 3.4 Expliquer comment supprimer un élément.

4 Nuages de points

On considère n points dans un espace métrique (les distances entre les points satisfont l'inégalité triangulaire). On veut partitionner les points en k groupes de manière à minimiser le plus grand diamètre d'un groupe. Le diamètre d'un groupe est la distance maximale entre deux points de ce groupe. Noter que n et k sont fixés dans l'énoncé du problème.

Question 4.1 On suppose que le diamètre optimal d est connu. Trouver une 2-approximation pour le problème.

Question 4.2 On considère l'algorithme qui, k fois, choisit comme "centre" le point à distance maximale de tous les centres déjà choisis, puis alloue chaque point au centre le plus proche. En faisant le lien avec la question précédente, montrer que cet algorithme est une 2-approximation pour le problème.

5 Ordonancement à une machine

On a n tâches indépendantes à exécuter sur une seule machine. La durée de la tâche T_j est p_j, son coût est w_j. Pour tout j, p_j et w_j sont des entiers positifs. Dans un ordonnancement, la première tâche commence son exécution au top 0, et la tâche T_j finit son exécution au top C_j. La fonction objective est la minimisation de la somme pondérée des temps de fin d'exécution, i.e. $S = \sum_{j=1}^{n} w_j C_j$

Question 5.1 Exemple avec $n = 4$, $(p_1, p_2, p_3, p_4) = (2, 3, 1, 4)$, $(w_1, w_2, w_3, w_4) = (2, 3, 1, 4)$. Que vaut S si on exécute les tâches dans l'ordre (T_1, T_2, T_3, T_4) ? Peut-on faire mieux ? Si oui, quel est l'optimal ?

Question 5.2 Dans le cas général, proposer un algorithme (polynomial) optimal.

On rajoute des temps de disponibilité : T_j ne peut pas commencer avant le top r_j.

Question 5.3 Traiter l'exemple précédent avec $(r_1, r_2, r_3, r_4) = (6, 2, 1, 0)$. Que vaut S si on exécute les tâches dans l'ordre (T_1, T_2, T_3, T_4) ? Peut-on faire mieux ?

Question 5.4 *(difficile)* On suppose $w_j = 1$ pour tout j. Quelle serait la stratégie optimale si on avait le droit de morceler l'exécution des tâches en plusieurs tranches de durée entière ?

Question 5.5 *(très difficile)* Toujours avec $w_j = 1$ pour tout j, en déduire un algorithme d'approximation de la solution optimale (on admettra que le problème est NP-complet).