Algorithmique I
Année 2008-2009

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

TD n°10
$\mathcal{NP}$-Complétude et Approximation

1 $\mathcal{NP}$-complétude de 2-Partition

On définit le problème de décision 2-Partition ainsi : soit $S = \{a_1, \ldots, a_n\}$ des entiers, existe-t-il $I \subset S$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$?

Question 1.1 Montrer que 2-Partition est NP-complet.

2 3-Partition à la 2-Partition

Étant donnés n entiers $S\{a_1, a_2, \ldots, a_n\}$, peut-on trouver trois sous-ensembles I_1, I_2 et I_3 partitionnant $[1..n]$ et tels que $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i = \sum_{i \in I_3} a_i$?

Question 2.1 Montrer que 3-Partition à la 2-Partition est NP-complet.

Question 2.2 S'agit-il de NP-complétude au sens faible ou au sens fort ?

3 Approximabilité de SUBSET-SUM

Dans TD le précédent, on s'est intéressé au problème de décision SUBSET-SUM consistant à savoir s'il existe un sous-ensemble d'entiers de S dont la somme vaut exactement t. Le problème d'optimisation qui lui est associé prend aussi en entrée un ensemble d'entiers strictement positifs S et un entier t, il consiste à trouver un sous-ensemble de S dont la somme est la plus grande possible sans dépasser t (cette somme qui doit donc approcher le plus possible t sera appelée *somme optimale*).

On suppose que $S = \{x_1, x_2, \ldots, x_n\}$ et que les ensembles sont manipulés sous forme de listes triées par ordre croissant. Pour une liste d'entiers L et un entier x, on note $L+x$ la liste d'entiers obtenue en ajoutant x à chaque entier de L. Pour les listes L et L', on note **Fusion**(L, L') la liste contenant l'union des éléments des deux listes.

Premier algorithme

début
$n \leftarrow |S|$;

$L_0 \leftarrow \{0\}$;

pour i **de** *1* **à** n **faire**
$L_i \leftarrow$ **Fusion**$(L_{i-1}, L_{i-1} + x_i)$;

Supprimer de L_i tout élément supérieur à t;

fin

retourner *le plus grand élément de* L_n;

fin

Algorithme 1 : Somme(S, t)

Question 3.1 Quelle est la distance entre la valeur retournée par cet algorithme et la somme optimale ?

Question 3.2 Quelle est la complexité de cet algorithme dans le cas général ? Et si pour un entier $k \geq 1$ fixé, on ne considère que les entrées telles que $t = \mathcal{O}(|S|^k)$, quelle est la complexité de cet algorithme ?

Deuxième algorithme Cet algorithme prend en entrée un paramètre ϵ en plus, où ϵ est un réel vérifiant $0 < \epsilon < 1$.

début
$n \leftarrow |S|$;

$L_0 \leftarrow \{0\}$;

pour i **de** *1* **à** n **faire**
$L_i \leftarrow$ **Fusion**$(L_{i-1}, L_{i-1} + x_i)$;

$L_i \leftarrow$ **Seuiller**$(L_i, \epsilon/2n)$;

Supprimer de L_i tout élément supérieur à t;

fin

retourner *le plus grand élément de* L_n;

fin

Algorithme 2 : Somme-avec-seuillage(S, t, ϵ)

L'opération **Seuiller** décrite ci-dessous réduit une liste $L =< y_0, y_1, \ldots, y_m >$ (supposée triée par ordre croissant) avec le seuil δ :

début
$m \leftarrow |L|$;

$L' \leftarrow < y_0 >$;

dernier $\leftarrow y_0$;

pour i **de** *1* **à** m **faire**
si $y_i > (1+\delta)$*dernier* **alors**
Insérer y_i à la fin de L';

dernier $\leftarrow y_i$;

fin

fin

retourner L';

fin

Algorithme 3 : Seuiller(L, δ)

Question 3.3 Évaluer le nombre d'éléments dans L_i à la fin de la boucle. En déduire la complexité totale de l'algorithme. Pour donner la qualité de l'approximation fournie par cet algorithme, borner le ratio valeur retournée/somme optimale.

4 Encore des problèmes $\mathcal{NP}$-complets

4.1 2-Partition-Approx

Question 4.1 Etant donnés n entiers $a_1, a_2, \ldots, a_n$, peut-on trouver un sous-ensemble $I \subset [1..n]$ tel que $|\sum_{i \in I} a_i - \sum_{i \notin I} a_i| \leq 1$

4.2 2-Partition avec même cardinal

Question 4.2 Soient $n = 2p$ un entier pair, et n entiers strictement positifs $a_1, a_2, \ldots, a_n$. Existe-t-il une partition de $\{1, 2, \ldots n\}$ en deux ensembles I et I' de même cardinal p et tels que $\sum_{i \in I} a_i = \sum_{i \in I'} a_i$?

4.3 Chevaliers de la table ronde

Question 4.3 Chevaliers de la table ronde

Étant donnés n chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

4.4 Vertex Cover avec degré pair

Question 4.4 Soient $G = (V, E)$ un graphe dont tous les sommets sont de degré pair, et $k \geq 1$ un entier. Existe-t-il un ensemble de sommets de G couvrant toutes les arêtes et de taille inférieure ou égale à k ?