ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 7 7 Nov. 2011

1.Sac à dos rationnel

a) Supposons que nous avons:

i) n objets de poids $w_1, w_2, \ldots, w_n$ et valeurs $v_1, v_2, \ldots, v_n$.

ii) Poids maximal $W \in \mathbb{N}$

La tâche est de trouver $x_i \in \mathbb{Q}$, $0 \leq x_i \leq 1$, tels que

$$\sum_{i=1}^{n} x_i w_i \leq W$$

et tels que $\sum_{i=1}^{n} x_i v_i$ est maximisé.

Montrer que l'algorithme glouton donné dans le cours est optimal.

b) Supposons qu'un vendeur dispose des produits suivants:

produit	quantité en grammes	valeur totale en $
A	200	1600
B	10	100
C	30	450
D	100	600

Il veut faire une sortie, en ne prenant pas plus que 50 grammes de produit, en maximisant la valeur totale de ce qu'il prend. Quelles quantités de chaque produit doit il prendre avec lui?

2.Sac à dos 0/1 avec des poids rationnels

Cet exercice suggère que le problème du sac à dos est peut-être plus difficile qu'on pourrait le croire.

a) Donner le temps de parcours de l'algorithme sac à dos 0/1 du cours en fonction de W et du nombre n d'objets candidats.

b) Pour le problème du sac à dos 0/1 comme vu dans le cours, on avait toujours supposé que les poids ainsi que les valeurs étaient des entiers. L'algorithme, s'applique-t-il encore si les valeurs sont des rationnels?

c) On aimerait maintenant modifier l'algorithme du sac à dos 0/1 pour qu'il fonctionne aussi avec des poids rationnels. Que faut-il changer pour pouvoir l'appliquer dans ce cas? (*Indication:* il suffit de multiplier W ainsi que les poids w_i par une valeur appropriée.)

d) Montrer qu'avec le changement du point précédent, il est possible de trouver des poids $w_i = p_i/q_i$, $i = 1, \ldots, n$ (où $p_i, q_i \in \mathbb{N}$) tels que le W de l'algorithme modifié sera alors exponentiel en fonction de

$$\mathrm{length}(w_1, \ldots, w_n) := \sum_{i=1}^{n} \log(p_i) + \log(q_i).$$

En déduire que l'algorithme aura alors un temps de parcours exponentiel en fonction de $\mathrm{length}(w_1, \ldots, w_n)$ dans le pire des cas.

e) Expliquer l'intérêt de la fonction $\mathrm{length}(w_1, \ldots, w_n)$.

En résumé donc, l'algorithme du sac à dos 0/1 classique adapté au cas des poids rationnels donne un algorithme exponentiel et donc inutile en pratique. Une autre idée est d'adapter l'algorithme glouton du sac à dos rationnel à ce problème. On considère l'algorithme suivant:

1. On trie les couples (v_i, w_i), $i = 1, \ldots, n$ de façon à ce que la suite $(v_i/w_i)_{i=1,\ldots,n}$ soit décroissante.

2. Commençant avec $i = 1$, on décide si on veut prendre l'objet i: s'il reste encore assez de place dans le sac, on le prend, sinon on ne le prend pas. On incrémente i et on itère le procédé de décision.

f) Montrer que cet algorithme termine en un temps polynomial en la taille de l'input.

g) Montrer que cet algorithme ne trouve pas toujours la solution optimale. (*Indication:* donner un exemple.)

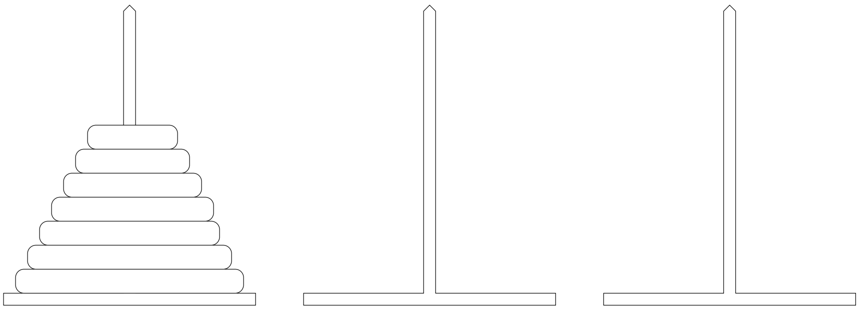
h) Construire des exemples pour lesquels l'algorithme trouve une solution arbitrairement loin de la solution optimale. Plus concrètement, on veut pour tout $\varepsilon > 0$ une instance telle que

$$\frac{X_{\mathrm{glout}}}{X_{\mathrm{opt}}} \leq \varepsilon,$$

où X_{glout} est la valeur de la solution trouvée de l'algorithme glouton, et X_{opt} est valeur de la solution optimale. (*Indication*: Il existe de tels constructions avec seulement deux objets.)

3. Les tours de Hanoi

On considère le problème suivant: On a n disques de différents diamètres qui se trouvent sur un bâton. Les disques sont ordonnés selon leur diamètre, avec le plus grand disque au fond. On dispose aussi de deux bâtons, initialement vides.



On aimerait déplacer la tour du premier bâton sur le dernier bâton. Le seul mouvement admis est d'enlever le plus haut élément d'une tour et de le rajouter sur un bâton vide ou un bâton dont le plus haut disque a un plus grand diamètre. Un disque ne peut donc jamais être posé sur un disque de plus petit diamètre.

a) Trouvez une procédure récursive qui donne la séquence des mouvements à effectuer pour déplacer une tour de taille n.

b) Calculer le nombre de mouvements qui sont effectués pour déplacer une tour de taille n. (Indication: Donner d'abord une formulation récursive, ensuite prouver la formule explicite par induction.)

c) Montrer que cet algorithme est optimal, i.e. qu'il n'en existe pas un meilleur. (Indication: Considérer l'étape de déplacement du plus grand disque. Qu'est-ce qui est arrivé avant et qu'est-ce qui reste nécessairement à faire?)