ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 7 9 Novembre 2009

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 (page 127) 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. Le codage de Huffman

Lesquels des codes suivants ne sont pas des codes de Huffman (pour aucune fréquence de lettres)?

a) $(0, 10, 11)$

b) $(00, 01, 10, 110)$

c) $(01, 10)$

3. Le code optimal n'est pas unique

Considérer l'alphabet (a, b, c, d) avec les fréquences $(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12})$.

a) Construire un code de Huffman pour cet alphabet avec les fréquences données.

b) Montrer que pour cet alphabet il existe deux codes de Huffman de longeurs $(1, 2, 3, 3)$ et $(2, 2, 2, 2)$.

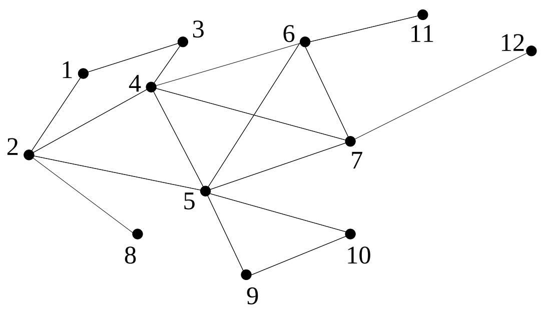
c) Pour chaque code calculer la longueur moyenne du code.

4. Problème de la clique

Une *clique* dans un graphe non orienté est un sous-graphe dans lequel chaque sommet est connecté à tous les autres sommets. Formellement, si $G = (V, E)$ est un graphe non-orienté, alors une clique de G est un sous-ensemble $W \subseteq V$ avec

$$\forall w_1, w_2 \in W \quad \text{avec } w_1 \neq w_2 \;:\; (w_1, w_2) \in E$$

a) Trouver 3 cliques de taille > 2 dans le graphe suivant:



Considérer l'algorithme glouton suivant pour trouver une clique dans un graphe donné:

Call: FINDCLIQUE(G)
Input: Graphe non-orienté connexe $G = (V, E)$
Output: Sous-ensemble $U \subseteq V$ pour lequel $\forall u_1, u_2 \in U$ avec $u_1 \neq u_2$: $(u_1, u_2) \in E$

$U \leftarrow V$
while U n'est pas une clique **do**
  choisir $u \in U$ pour lequel il existe un $u' \in U$ qui n'est pas connecté à u
  $U \leftarrow U \backslash \{u\}$
return U

b) Le problème de trouver *la clique la plus grande* d'un graphe est en général très difficile. Cet algorithme ne le résout pas.

Donner un exemple d'un graphe connexe, et d'un parcours de cet algorithme sur ce graphe qui ne retourne pas la clique la plus grande de ce graphe.

c) Montrer que pour tout $s > 2$ il existe un graphe connexe dont la clique la plus grande est de taille s mais pour lequel il y a un parcours de cet algorithme qui retourne une clique de taille ≤ 2.