Prof. F. Eisenbrand EPFL - DISOPT

Optimisation Discrète

Semestre de printemps 2013

Série 10 **16 mai 2013**

Remarque générale :
Si vous voulez obtenir un bonus pour l'évaluation finale, vous pouvez rendre des solutions écrites aux exercices notés, au plus tard le **lundi 27 mai**, avant 12h dans la boîte au bureau MA B1 533.
Le rendu peut être fait en groupe de trois personnes au plus.

Exercice 1 ($*$)
Deux joueurs A et B conviennent de participer au jeu suivant :
De façon alternée, ils lancent en l'air une pièce équilibrée ; le côté face rapporte un point à A, le coté pile à B. Le premier à n points remporte la victoire.
Considérer le problème suivant : Donné deux nombres entiers $a, b \in \mathbb{N}$, quelle est la probabilité que le joueur A gagne si A et B ont déjà a et b points ?
Trouver un algorithme qui résout le problème en temps $O(n^2)$. La déscription de l'algorithme doit comporter :

(a) Une déscription en mots d'une entrée du tableau du programme dynamique
(b) Une déscription (en langage mathématique ou quelques phrases compréhensibles) comment on peut calculer une telle entrée
(c) Un argument de correction

Exercice 2

Soit $D = (V, A)$ un graphe orienté avec poids $w : A \to \mathbb{R}_{\geq 0}$.

(i) Soient $k \in \mathbb{N}$ et $s, w \in V$ deux sommets. De plus, soit P le chemin le plus court de s à w en utilisant $\leq k$ arcs et soit $a = (v, w)$ l'arc final de ce chemin.
Démontrer que $P \setminus \{a\}$ est un plus court chemin de s à v.
(ii) Décrire un algorithme qui trouve les chemins les plus courts depuis un sommet $s \in V$ à l'aide de la programmation dynamique.

Exercice 3
Le problème du rendu de monnaie est le suivant : étant donné un système de monnaie avec n pièces différentes, comment rendre une somme donnée M de façon optimale, c'est-à-dire avec le nombre minimal de pièces et billets ?
Donner un algorithme qui résout ce problème avec l'aide de la programmation dynamique.

Exercice 4

Considérer le problème du sac à dos.

Étant donné n objets avec poids $w_1, \ldots, w_n$ et valeurs $c_1, \ldots, c_n$, il s'agit de choisir un sous-ensemble des objets qui maximise la valeur totale sans dépasser le poids maximum (ou la capacité) W du sac à dos.

Trouver un algorithme qui le résout en temps $O(n \cdot W)$.