Licence Info
Algorithmique Avancée
TD 2: Octobre 2006

1. On considère l'algorithme glouton pour le problème du sac-à-dos discret, qui consiste à regarder tous les objets dans leur ordre de *valeur/poids* et de mettre un objet dans le sac si son poids ne dépasse pas la capacité restante.

   - Est-il vrai que cette méthode est garantie de trouver une solution avec valeur totale au moins 50% de celle de la solution optimale?

     Réponse: Non. Contre-exemple: capacité 10, deux objets (dans l'ordre) poids 1, valeur 2 et poids 10 valeur 10.
   - Quel est le plus grand c pour lequel cette méthode est garantie de trouver une solution avec valeur totale au moins c fois celle de la solution optimale?

     Réponse: 0. Pour n'importe quel $c > 0$, disons $c > 1/N$, on peut donner un contre-exemple pareil:
     capacité $2N$, deux objets: poids 2 valeur 2 et poids $2N - 1$ valeur $2N$.

2. Dessiner l'arbre engendré par l'insertion dans un arbre binaire de recherche (vide au départ) de:

   (a) 10, 15, 8, 20, 30, 12
   Réponse: (8) 10 ((12) 15 (20 (30)))

   (b) 8, 10, 15, 20, 30, 12
   Réponse: 8 (10 ((12) 15 (20 (30))))

   (c) 10, 15, 20, 30, 8, 12
   Réponse: (8) 10 ((12) 15 (20 (30)))

3. (a) Combien y a-t-il de formes possibles d'un arbre binaire; de 2 sommets, de 3 sommets, de 4 sommets, de 5 sommets?
   Réponses: 2, 5, 14, 42.

   (b) Si j'insère les entiers 1, 2, 3, 4, 5 (dans un arbre binaire de recherche vide au départ) dans un ordre aléatoire, montrer un arbre avec probabilité 1/120 d'être engendré et un autre avec probabilité 1/15; quelles sont les autres probabilités possibles pour d'autres arbres?
   Réponse: 1/120: une chaîne, par exemple 1(2(3(4(5))));
   1/15: (1) 2 ((3) 4 (5));
   dans tous les cas la probabilité est le réciprocal du produit des tailles de tous les sous-arbres non-vides (y compris l'arbre lui-même);
   autres possibilités: 1/20, 1/30, 1/40, 1/60

(c) Selon les mathématiciens la hauteur moyenne d'un arbre binaire de n sommets est $\Omega(\sqrt{n})$ mais selon le cours elle est $O(log\ n)$; expliquez la contradiction.
Réponse: le $\Omega(\sqrt{n})$ est pour le cas où tous les arbres ont la même probabilité; le processus d'insertion dans les arbres de recherche donne une loi de probabilité non-uniforme (comme remarqué dans la réponse précédente) et tend à favoriser les arbres plus équilibrés et moins hauts.

4. (a) Donner du code pour les deux types de rotations utilisées pour rééquilibrer des arbres AVL. On supposera que P est une variable de type `pointeur` qui pointe vers la racine de l'arbre où le rééquilibrage est nécesaire et que le sous-arbre rendu trop haut par l'insertion est celui de droite (l'image miroir des exemples vus en cours.) **Réponse** On suppose que les deux pointeurs dans un noeud sont *gauche* et *droite* et qu'une autre variable de type `pointeur` est disponible:

   i.
```
temp:=P^.droite;
P^.droite:=temp^.gauche;
temp^.gauche:=P;
P:=temp;
```
   ii.
```
temp:=P^.droite^.gauche;
P^.droite^.gauche:=temp^.droite;
temp^.droite:=P^.droite;
P^.droite:=temp^.gauche;
temp^.gauche:=P;
P:=temp;
```

   (b) Si la hauteur de l'arbre était h avant la rotation quelles sont les hauteurs possibles après? **Réponse** Une seule est possible, h, (ce qui garantit qu'après une rotation dans un sous-arbre l'arbre entier reste équilibré.)