TD 7

Arbres

Exercice 1 On rappelle que la hauteur d'un arbre peut être définie comme le maximum des profondeurs de ses noeuds et que la profondeur d'un noeud k est définie par :

- $p(k) = 0$, si k est la racine
- $p(k) = p(k') + 1$, si k est l'un des fils de k'

On suppose qu'un arbre est représenté par un tableau $PERE$ (où $PERE[i] = j$ indique que le père du noeud i est le noeud j). Par convention, pour la racine r de l'arbre, on a $PERE[r] = r$.

1. On suppose initialement que les noeuds sont numérotés de façon telle que le père de chaque noeud i porte un numéro plus petit que i. Donner (et justifier) un algorithme en $\Theta(n)$ qui calcule la hauteur de l'arbre.
2. On relâche maintenant l'hypothèse sur la numérotation des noeuds (le père d'un noeud i peut porter un numéro plus grand ou plus petit que i). Peut-on toujours trouver un algorithme en $\Theta(n)$ qui calcule la hauteur de l'arbre ? Dans l'affirmative, donner cet algorithme et justifier qu'il a bien une complexité linéaire ; sinon, expliquer pourquoi ceci n'est pas faisable, donner l'algorithme le plus rapide possible qui effectue le calcul de la hauteur et donner sa complexité.

Exercice 2 Un tas est un arbre binaire satisfaisant deux propriétés. La première (dite ici *Propriété 1*) concerne la disposition des nœuds dans l'arbre (la "forme" de l'arbre), la deuxième concerne l'ordre relatif des éléments (ou plutôt de leurs clés) stockés dans les nœuds de l'arbre. La Propriété 1, en particulier, permet de représenter l'arbre sous forme de tableau.

1. Donner la représentation sous forme d'arbre du tableau $T = [15; 27; 31; 19; 53; 73; 14]$ en sachant qu'il s'agit d'un arbre binaire satisfaisant la Propriété 1.
2. En supposant que vous disposez d'une méthode `entasser(int [] X, int i)` qui entasse l'indice i dans un tas X (plus précisément : si les arbres ayant comme racine les fils du nœud i sont des tas max, alors `entasser(int [] X, int i)` transforme le sous-arbre de racine i en un tas max), rappelez l'algorithme qui permet de transformer en tas max un tableau quelconque représentant un arbre satisfaisant la Propriété 1.
3. Donner les représentations des arbres intermédiaires obtenus en appliquant cette construction au tableau T.
4. On considère à présent le tas max TM suivant : $[100; 50; 20; 17; 35; 5; 19]$. Donnez lez 7 arbres intermédiaires obtenus pendant la procédure de tri par tas appliquée à TM.

Exercice 3 On peut trier un ensemble donné de n nombres en commençant par construire par insertions successives un arbre binaire de recherche (initialement vide) contenant ces nombres, puis en effectuant un parcours infixe de l'arbre. Quels sont les temps d'exécution de cet algorithme dans le pire et dans le meilleur des cas ? Justifiez vos réponses. Dans quelle situation on peut rencontrer le pire des cas ?

Exercice 4 1. Construire par insertions successives de $3, 5, 1, 6, 8, 11, 2$, en précisant les arbres obtenus à chaque étape :

- le tas correspondant
- l'ABR correspondant
- l'ARN correspondant

2. On rappelle qu'un arbre binaire de hauteur h est dit *complet* si pour tout $p \leq h$ il possède exactement 2^p noeuds de profondeur p. Dessinez un arbre complet dont les noeuds contiennent les valeurs $1, 2, 3, 4, 5, 6, 7$ et qui soit un ABR. Effectuez une rotation gauche sur la racine, puis une rotation droite sur le noeud étiqueté par 4. Donnez alors une coloration pour obtenir un arbre rouge noir.
3. Quel est le nombre maximum de noeuds dans un ARN de hauteur noire h ?

Exercice 5 On rappelle que dans un AVL chaque nœud contient un champs *bal* (pour balance) dont la valeur marque la différence de hauteur entre les sous-arbres du noeud.

1. Dessinez la suite des arbres AVL construits par insertions successives des entiers $5, 6, 7, 2, 1, 3, 4$ dans un AVL initialement vide, en expliquant les étapes des insertions. On souhaite voir précisément le résultat de l'insertion dans l'arbre avant les rotations, les valeurs des balances, la suite des rotations qui mène à l'AVL.
2. Dessinez l'arbre AVL complet dont les nœuds sont $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$. Dans cet AVL donner le résultat et les étapes intermédiaires de la suppression des valeurs : $5, 6, 7, 1$