

EA4 – Examen du 30 Mai 2012 - durée : 2h30

La qualité de la rédaction sera prise en compte dans la notation. Justifiez toutes vos réponses et expliquez les fondements de vos algorithmes en français avant de les rédiger en pseudo-code compréhensible et commenté. Le barème est donné seulement à titre indicatif.

Les algorithmes donnés sans commentaires ou sans explications ne seront pas pris en compte par le correcteur. Documents non autorisés.

Exercice 1 (4 points).

Soit la fonction :

```
int f(int n) {
    int m = abs(n); int res = 0;
    tantque (m>1) faire
        si (m est pair) alors m:=m/2 sinon m:=m-8;
        res=res+1;
    ftq;
    retourne(res);
}
```

1. Quel est le temps de calcul de f relativement à n dans le meilleur cas ? (Donner la quantité exacte, pas le Grand-O)
2. Quel est le temps de calcul de f relativement à n dans le pire des cas ? (Donner la quantité exacte, pas le Grand-O)
3. Donner une valeur de n qui réalise le meilleur cas et une qui réalise le pire des cas.
4. Quelle est la probabilité d'avoir le meilleur cas ?
5. Et celle d'avoir le pire des cas ?

Exercice 2 (10 points).

Les fonctions demandées dans cet exercice prennent en paramètre un arbre binaire donné par l'adresse de sa racine, qui est un objet de type :

```
struct noeud {int val; struct noeud * g; struct noeud * d};
```

Elles doivent toutes avoir une complexité en temps en $O(N)$ où N est le nombre de noeuds de l'arbre. Vous expliquerez les raisonnements derrière vos algorithmes et vous justifierez leur complexité. Veuillez noter que des solutions élégantes et succinctes existent, qui comprennent environ dix lignes de pseudo-code pour les deux premières questions, et moins de vingt pour la troisième.

1. Une fonction booléenne qui retourne VRAI si et seulement dans l'arbre binaire donné tout noeud interne a exactement deux fils. (3 points)
2. Une fonction qui, pour un entier h , retourne h si dans l'arbre binaire donné tout noeud interne a exactement deux fils et toutes les feuilles se trouvent à la même profondeur h . Retourne -2 sinon. (Pourquoi on ne fait pas retourner -1 ?) (3 points)
3. Une fonction qui, pour un entier h , retourne h si l'arbre binaire donné a la forme d'un tas de hauteur h (tous les niveaux sont "complets", sauf éventuellement le dernier de profondeur h , dans lequel les noeuds "occupés" sont ceux les plus à gauche). Retourne -2 sinon. (4 points)

Exercice 3 (6 points).

Proposer un algorithme pour trier des valeurs entières stockées dans un tableau qui utilise comme structure auxiliaire un ABR. Les fonctions de gestion des ABR ne devront pas être réécrites.

1. Quel est le pire des cas de votre algorithme et quelle est sa complexité en temps ?
2. Quelle est la complexité en temps du meilleur des cas de votre algorithme ?
3. Quelle complexité en moyenne vous attendez pour votre algorithme ? Pourquoi ?
4. Que peut-on dire des complexités si on utilise un AVL à la place d'un ABR ?