Université de Versailles St Quentin — 05 juin 2023
Master 1 AMIS, DataScale, IRS, SeCReTS — Durée: 1h30

Compléments d'algorithmique et complexité
Examen

Une unique feuille A4 recto-verso est autorisée avec ce que vous voulez écrit dessus. Aucun livre autorisé. Aucun appareil électronique autorisé.

Votre copie doit être claire et bien rédigée. Vos réponses aux questions doivent être justifiées. Le barème est indicatif. Le sujet est recto-verso

---

Exercice 1 – Des arbres .... *(5 points)*

1. Ajouter la valeur du dernier chiffre de votre numéro étudiant à chacun des éléments de la suite 12,1,5,7,3,9,4,2.
2. Considérons cette suite modifiée comme le résultat d'une liste issue d'un parcours en largeur d'un arbre binaire complet. Pour simplifier on considérera que sur le dernier niveau les feuilles se trouvent le plus à gauche possible, comme dans un tas. Dessiner l'arbre binaire complet dont est issue cette liste.
3. Donner la suite de nombres obtenue en faisant un parcours en profondeur postfixe de l'arbre de la question précédente.
4. Insérez les éléments de la suite de la question précédente, un à un dans un arbre binaire de recherche. Vous représenterez chacune des étapes.
5. Supprimer la racine de cet arbre binaire de recherche de manière à obtenir un nouvel arbre binaire de recherche. Représentez ce nouvel arbre et expliquez comment vous l'avez obtenu.

Exercice 2 – Encore des arbres .... *(5 points)*

Etant donnée la structure de données suivante vue en cours :

```
Enregistrement Nœud {
Val     : entier;
Gauche : ↑ Nœud;
Droit : ↑ Nœud;
}
```

1. Donner un algorithme **récursif** qui ajoute 2 à la valeur des sommets internes d'un arbre binaire et 1 à ses feuilles. Quelle est sa complexité ?
2. Si on applique l'algorithme précédent à un ABR, est-ce toujours un ABR ? Pourquoi ?

Exercice 3 – Des listes ... *(4 points)*

Etant donnée la structure de données suivante :

```
Enregistrement Element {
Val     : entier;
Suivant : ↑ Element;
}
```

1. Donner un algorithme récursif qui prend en entrée une liste l et un entier x et qui renvoie le nombre d'occurrences (répétitions) de x dans l.
2. Donner un algorithme récursif qui calcule le produit des valeurs d'une liste donnée en entrée.

Exercice 4 – Des permutations ..... *(6 points)*

Une permutation p sur $1..n$ est inférieure selon l'ordre lexicographique à une permutation q s'il existe $1 \leq i < n$ tel que $p[1] = q[1], p[2] = q[2], \ldots, p[i-1] = q[i-1]$, et $p[i] < q[i]$. Par exemple, si $n = 5$, la permutation $p : 1; 3; 5; 4; 2$ (c'est-à-dire $p[1] = 1, p[2] = 3, p[3] = 5, p[4] = 4, p[5] = 2$) est inférieure à la permutation $q : 1; 5; 3; 4; 2$ car $p[2] < q[2]$.

1. Écrire les 10 plus petites permutations dans l'ordre lexicographique pour $n = 5$.
2. Écrire un algorithme qui étant données deux permutations p et q (données sous forme de tableaux d'entiers) renvoie vrai si $p < q$. Vous donnerez la complexité de votre algorithme.
3. Écrire un algorithme qui, étant donnée une permutation p, trouve le plus grand entier i tel que $p[i] < p[i+1]$, et retourne -1 si ce i n'existe pas.