Université de Versailles Saint-Quentin — Structure de données et algorithmique
L2 — Année 2014-2015

Examen

mercredi 17 juin 2015

Toute réponse donnée doit être **justifiée** et tout algorithme doit être **expliqué** au moins succinctement. Le barème est indicatif. L'examen dure deux heures et les notes prises en cours et td sont autorisées. Tout autre document, ordinateur, téléphone ... doit être rangé.

Exercice 1 L'arbre qui cache l'examen (4 points)

1. Construisez un arbre binaire de recherche en insérant successivement 6,4,16,10,1,3,88,14,19,100. Vous représenterez chacune des étapes.
2. Supprimez les éléments 10 puis 88.
3. Choisissez un noeud déséquilibré auquel vous appliquerez des rotations afin de le rééquilibrer. Expliciter les rotations.

Exercice 2 Manipulation de liste (6 points)

Nous utiliserons une liste *doublement chaînée*. Le type liste est :

```
Enregistrement Liste {
   début : ↑Element;
   fin : ↑Element;
}
```

et d'un élément :

```
Enregistrement Element {
   valeur : entier;
   précédent : ↑Element;
   suivant : ↑Element;
}
```

Pour chaque question **donner la complexité** de votre algorithme en fonction de la taille de la ou des listes.

1. Donner une procédure qui insère un élément au début de la liste.
2. Donner une procédure qui concatène deux listes.
3. Donner une procédure qui inverse l'ordre des éléments d'une liste.
4. Donner une procédure qui teste si une liste est un palindrome.

Exercice 3 Arbres arithmétique (6 points)

Un arbre arithmétique est défini de la manière suivante :
— c'est un arbre binaire
— chaque feuille de l'arbre est étiquetée (champ valeur) par un entier

1. Écrire une procédure qui renvoie la somme des valeurs des feuilles d'un arbre donné.
2. Écrire une procédure qui prend en entrée un noeud interne v et qui le remplace par une feuille dont la valeur est la somme des valeurs des feuilles du sous-arbre.

3. Les sommets ont également un champ type : si c'est une feuille il vaut 0, si c'est un noeud interne qui représente une somme il vaut 1 et si le noeud représente une multiplication il vaut 2. Un noeud v qui a deux fils v_1 et v_2 vérifie $v.valeur = v_1.valeur + v_2.valeur$ si v est de type 1 et $v.valeur = v_1.valeur * v_2.valeur$ si v est de type 2.

   Donner une procédure itérative **et** une procédure récursive qui prend en entrée un arbre dont les champs valeurs des noeuds internes ne sont pas initialisés et qui les calcule.

Exercice 4 Files (4 points)

Vous disposez des 4 fonctions suivantes pour manipuler des piles :

`créer_nouvelle_file(), file_vide(p), défiler(p) et enfiler(p,e)`

où p est une pile et e un élément. Ces fonctions ont le même comportement que celles vues en cours et en td. Vous ne connaissez pas l'implémentation précise de ces piles et ne pouvez donc utiliser que ces fonctions pour les manipuler.

1. Donner un algorithme qui calcule le minimum d'une file.
2. Donner un algorithme qui étant donné deux entiers i et j et une file f, modifie f de façon à ce que l'élément en position i se retrouve en position j (sans que les autres éléments ne changent de position). Vous n'utiliserez pour ça que la file f et des variables locales.
3. Comment, à partir de l'algorithme précédent, peut-on retourner une file ?