

EA4 – Examen du 22 Juin 2011 - 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 où nécessaire. 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 E un ensemble contenant n entiers. Décrire un algorithme efficace pour déterminer s'il y a deux (ou plus) entiers identiques dans E et donner sa complexité en temps.

Note : on peut, bien évidemment, résoudre le problème avec un algorithme trivial en temps $\Theta(n^2)$ dans le pire des cas, toutefois la solution permettant d'obtenir les points du barème doit être strictement meilleure que $\Theta(n^2)$ en temps (et $\Theta(1)$ en espace) (une solution en $\Theta(n^2)$ ne sera évalué que 1 point, si correctement implémentée).

Exercice 2 (4 points).

Soient A et B deux suites d'entiers, les deux contenant k éléments. Étant donné un entier x , donner un algorithme en $O(k \log k)$ en temps et $O(1)$ en espace pour déterminer s'il existe un entier a dans A et un entier b dans B tels que $x = a + b$.

Exercice 3 (6 points).

Considérer des arbres d'**arité quelconque**, basés sur une structure nœud de type suivant :

```
struct noeud {int val; struct noeud* pf; struct noeud* fd};
```

où **pf** et **fd** sont des pointeurs **vers le premier fils et vers le frère droit**, respectivement. L'absence d'un sous-arbre ou d'un frère droit est indiquée par le pointeur **NULL**.

On a vu en cours des procédures récursives de parcours d'arbre (infixe, préfixe, postfixe). Ici on se propose de parcourir l'arbre **par niveaux** (on parcourt d'abord la racine, puis les nœuds de profondeur 1 de gauche vers droite, puis ceux de profondeur 2 et ainsi de suite) et de le faire de manière **non récursive (itérative)**.

Écrivez un algorithme non récursif qui associe au champ **val** de chaque nœud la position à laquelle il sera rencontré pendant un parcours par niveaux. L'espace mémoire maximal autorisé est $O(n)$, vous pouvez donc utiliser des structures auxiliaires dont vous détaillerez les attributs et les méthodes.

Quel est la complexité en temps de votre algorithme ?

Exercice 4 (6 points).

Construire le tas-min T obtenu par la procédure suivante.

Étape 1. Stocker dans un **AVL** A les entiers

47, 53, 88, 24, 66, 91, 75, 9, 34, 11

Étape 2. Sortir ensuite les éléments un par un de l'AVL A en choisissant toujours celui à la racine et les insérer un par un dans le tas-min T .

Puisqu'au moins deux stratégies sont possibles pour supprimer un nœud d'un ABR (on rappelle que un AVL est un cas particulier d'ABR), indiquez celle que votre algorithme applique. Détaillez toutes les étapes avec des dessins et en particulier, expliquez les opérations de rééquilibrage effectuées à chaque fois en fonction du déséquilibre de l'arbre.