

Algorithmique et Structures de Données

Examen écrit (1h30)

G1: pascal.monasse(at)enpc.fr G2: thomas.belos(at)enpc.fr
G3: nicolas.audebert(at)lecnam.net G4: clement.riu(at)enpc.fr

29/03/2021

Les exercices sont indépendants. Il n'est pas interdit d'utiliser votre portable pour tester vos algorithmes, mais évidemment pas le wifi.

1 Encore un algorithme de tri

1. Étant donnés deux tableaux *triés* d'entiers $T1$ et $T2$ de taille de taille respective $N1$ et $N2$, proposer sous forme de code C++ leur fusion dans un tableau trié T (de taille donc $N = N1 + N2$).
2. Quelle est sa complexité en fonction de N ?
3. Proposer suivant ce principe un algorithme triant un tableau de 2^n entiers.
4. L'écrire sous forme de code C++ (faisant appel à la fonction `fusion` de la première question).
5. Quelle est sa complexité ?
6. Commenter cette complexité.

2 Arbre binaire de recherche

1. Rappeler la propriété vérifiée pour tout noeud d'un arbre représentant une file de priorité.
2. Quelle est la complexité pour trouver le plus grand entier dans une file de priorité ? Le plus petit entier ?
3. On appelle arbre binaire (au maximum deux enfants par noeud) de recherche (ABR) un arbre ayant la propriété que chaque noeud ait une valeur au moins aussi grande que tout noeud du sous-arbre gauche (s'il existe) et pas plus grande qu'aucun noeud du sous-arbre droit (s'il existe). Dessiner deux ABR différents contenant les entiers de 1 à 7, dont l'un complet (les feuilles sont toutes à la même profondeur, les autres noeuds ont tous deux enfants).
4. Montrer que si $N = 2^n - 1$ ($n \geq 1$) il existe un ABR complet contenant N entiers.
5. Proposer un algorithme pour trouver le plus petit élément d'un ABR et un autre pour trouver le plus grand.
6. Quelle est leur complexité en fonction de N si l'ABR est complet ?
7. Proposer un algorithme pour trouver si un entier est présent dans l'ABR.
8. Quelle est sa complexité pour un ABR complet ? La complexité la pire si l'ABR n'est pas complet ?
9. Proposer un algorithme pour trouver le prédécesseur ou le successeur immédiat de la valeur d'un noeud.
10. Proposer un algorithme pour la fonction `push(i)` qui insère la valeur i dans un ABR.
11. Complexité de l'opération `push` ?
12. Pour `pop(i)`, qui supprime la valeur i (supposée présente une seule fois) d'un ABR, on commence par trouver le noeud. Comment faire si ce noeud a 0 ou 1 enfant ?
13. Si le noeud a deux enfants, comment faire le `pop` ? (utiliser le prédécesseur ou successeur immédiat)
14. Finalement, quelle est la complexité du `pop` ?