Programmation avancée

Correction d'examen

Jean-Michel Begon

21 décembre 2018

1 Vrai ou faux

PA Sept. 2015 $7^{\log_2 N}$ est $O(N^3 + N^2)$.

PA Sept. 2015 Pour un ensemble donné de valeurs, on ne peut construire qu'un seul et unique tas-max.

Janvier 2015 Pour toutes fonctions croissantes et positives $f(n)$, $g(n)$ et $h(n)$, si $f(n) = O(g(n))$ et $f(n) = \Omega(h(n))$, alors $g(n) + h(n) = \Omega(f(n))$.

SDA Aout 2015 Il est impossible de construire un arbre binaire de recherche à partir d'un tableau quelconque contenant N clés en $\Theta(N)$ opérations.

PA Janv. 2015 Si le facteur de charge d'une table de hachage est plus petit que 1, alors il n'y a pas de collisions.

2 Analyse de complexité (Aout 2015)

Soit l'algorithme suivant :

```
Algo(n)
if n == 1
    return 0
j = 0
i = 1
while i < n
    j = j + 1
    i = 2i
return 2 * (Algo(n/2) + Algo(n/2) + Algo(n/2)) + 2^j
```

En supposant que la fonction prend en entré une puissance de 2 :

1. Donnez une forme analytique équivalente à cette fonction
2. Donnez une borne assymptotique sur la complexité de la fonction. *Suggestion :* utilisez le *Master theorem.*

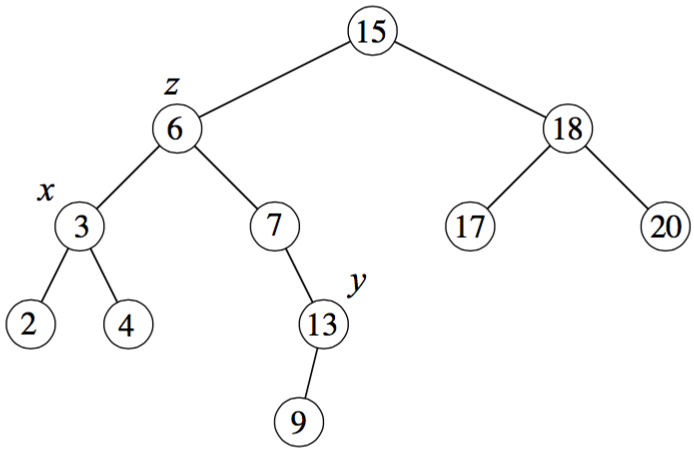
3 Arbre binaire de recherche (Janvier 2015)

1. Qu'est-ce qu'un arbre binaire de recherche ?
2. Le parcours préfixe d'un arbre binaire de recherche donne la s séquence suivante :

$$5, 1, 2, 11, 8, 7, 13, 12.$$

Dessinez un arbre correspondant à ce parcours. Cet arbre est-il unique ?

3. Ecrivez une fonction $getCCA(T, x, y)$ renvoyant l'ancêtre commun le plus proche des deux nœuds x et y dans un arbre binaire de recherche T. L'ancêtre commun le plus proche de deux nœuds x et y est l'ancêtre z de x et y qui est le plus profond dans l'arbre. Pour cette définition, on considérera un nœud comme un ancêtre de lui-même. Par exemple, pour l'arbre ci-dessous, les deux appels $getCCA(T, x, y)$ et $getCCA(T, z, y)$ doivent renvoyer le nœud z. (suggestion : tirez profit de la propriété d'arbre binaire de recherche)



4. Analysez la complexité de votre algorithme au pire et au meilleur cas.

4 Résolution de problèmes (Septembre 2012)

soit un tableau $A[1, n]$ de n entiers, chacun pris dans l'intervalle $[1, k]$. On chercher à déterminer s'il existe un sous-ensemble des entiers qui somme exactement à $S/2$, où S est la somme de tous les entiers de la liste.

1. Ecrivez un algorithme efficace pour résoudre ce problème en utilisant la programmation dynamique. *Suggestion : définissez une fonction $f(i, y)$ valant 1 s'il existe un sous-ensemble des i premiers entiers du tableau qui somme exactement à y, 0 sinon. Il s'agit alors de calculer efficacement $f(n, S/2)$.*
2. Analysez la complexité de cet algorithme de fonction de n et k.

5 Résolution de problèmes (Janvier 2015)

A partir d'un nombre n, on génère une séquence en enlevant un chiffre soit au début, soit à la fin de ce nombre, jusqu'à ce qu'il ne reste plus qu'un seul chiffe. La profondeur carrée $S(n)$ de n est définie comme le nombre maximum de carrés parfaits qu'on peut obtenir au long d'une séquence à partir de n (y compris n). Par exemple, $S(32492) = 3$ car :

$$32492 \rightarrow \mathbf{3249} \rightarrow \mathbf{324} \rightarrow 24 \rightarrow \mathbf{4}$$

où 3249, 324, et 4 sont des carrés parfaits et il n'y a pas d'autres séquences partant de 32492 contenant plus de carrés parfaits.

1. Décrivez un algorithme efficace basé sur la programmation dynamqiue pour calculer la profondeur $S(n)$ d'un nombre n de d chiffres $a_1a_2 \ldots a_d$. Précisez les équations de récurrences correspondant à votre solution (en ce compris le(s) cas de base). Analysez la complexité de votre algorithme au pire et au meilleur cas.