

Structures de données et algorithmes

Correction d'examen

Jean-Michel BEGON

17 mai 2019

1 Structure

L'examen se compose en général comme suit :

1. Vrai ou faux (et justification) sur toute la matière.
2. Le tri
 - Question théorique (pseudo-code, complexité, stabilité, etc.).
 - Question pratique : modification de l'algorithme.
3. Structures de données (arbre, tas, file à priorités, dictionnaires, union-find, etc.)
4. Graphe
 - Question théorique : pseudo-code et illustration d'un algorithme.
 - Question pratique : modification de l'algorithme.
5. Résolution de problème (programmation dynamique)

2 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.

3 Tri (Mai 2014)

1. Décrivez l'algorithme de tri rapide dans le formalisme de votre choix.
2. Implémentez une fonction `SELECT(A, k)` permettant de trouver la k -ième plus petite valeur d'un tableau d'entiers A , sans qu'il soit nécessaire de trier A . Cette fonction peut modifier l'ordre des éléments du tableau A .
Suggestion : Inspirez vous de l'algorithme du tri rapide.
3. Analysez la complexité de votre algorithme dans le pire et le meilleur cas. Justifiez.
4. Que pensez-vous de la complexité du cas moyen ?

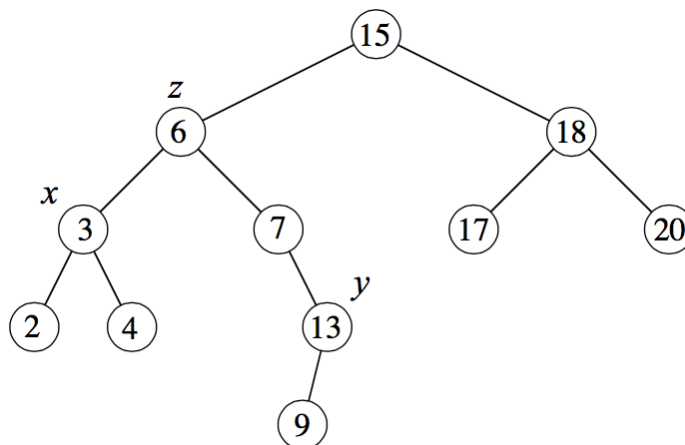
4 Arbre binaire de recherche (Janvier 2015)

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

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 chiffre. 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 :

$$32492 \rightarrow 3249 \rightarrow 324 \rightarrow 24 \rightarrow 4$$

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 dynamique pour calculer la profondeur $S(n)$ d'un nombre n de d chiffres $a_1 a_2 \dots a_d$. 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.