

Quick 2 – 19 novembre 2020 – durée 1h

Les documents et les téléphones (incluant smartphone, tablettes,... tout ce qui contient une interface réseau) sont interdits. Les calculettes sont autorisées.

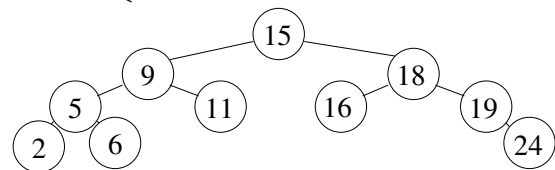
Seuls les dictionnaires pour les personnes de langue étrangère sont autorisés.

Toutes les réponses doivent être justifiées. Le barème est indicatif.

k -ième élément dans un Arbre Binaire de Recherche

Étant donné un arbre binaire de recherche A comportant n éléments, on cherche à déterminer le k -ième élément (dans l'ordre croissant), pour un entier $k \leq n$.

Par exemple, dans l'arbre ci-contre, le 3^e élément est 6 et le 7^e élément est 16.



Dans tout ce problème on notera h la hauteur de l'arbre A .

Recherche directe

Question 1 : (2 points)

Proposer un algorithme permettant de résoudre ce problème en $\mathcal{O}(n)$.
Vous justifierez rapidement sa complexité.

Question 2 : (2 points)

Donner également la complexité de cet algorithme en fonction de h et de k et justifier votre réponse.

Adaptation de la structure de données

Afin d'améliorer le résultat précédent, on propose d'effectuer un prétraitement sur la structure de données. Ainsi, chaque nœud a de l'arbre A sera marqué par le nombre de nœuds du sous-arbre dont il est la racine. On notera $taille(a)$ ce nombre, que ce soit pour le consulter ou pour le mettre à jour.

Question 3 : Le prétraitement (3 points)

Écrire un algorithme **efficace** prenant en argument un arbre binaire de recherche A non encore marqué, et permettant, par effet de bord, de marquer tous les nœuds de A avec leur taille.

Déterminer la complexité de cet algorithme.

Question 4 : Utilisation du prétraitement (5 points)

En supposant que les tailles des nœuds sont déjà calculées, écrire un algorithme qui profite de cette information pour rechercher le k -ième élément en temps $\mathcal{O}(h)$.

Encore une fois vous justifierez cette complexité.

Question 5 : (1 point)

Ce nouvel algorithme constitue-t-il une amélioration par rapport à celui de la question 1 ? Expliquer votre réponse.

Question 6 : Mise à jour d'un arbre marqué (4 points)

On suppose maintenant disposer d'un arbre binaire de recherche A déjà marqué par les tailles des sous-arbres.

Écrire un algorithme qui insère dans cet arbre un nouvel élément x à une position convenable, tout en mettant à jour les tailles des différents nœuds de l'arbre pour lesquels cela est nécessaire.

Cet algorithme ne devra pas être plus coûteux que l'insertion dans un arbre binaire de recherche « standard » (mais il n'est pas demandé de chercher à équilibrer l'arbre).

Une fausse bonne idée

Une idée similaire serait de marquer chaque nœud a de l'arbre A par son rang (autrement dit l'entier k qui donne sa position parmi les éléments de l'arbre rangés dans l'ordre).

Question 7 : (2 points)

En supposant que les rangs des nœuds sont déjà calculés, écrire un troisième algorithme de recherche du k -ième élément qui profite de cette information pour obtenir une complexité en $\mathcal{O}(h)$.

Question 8 : (1 point)

Expliquer rapidement pourquoi cette solution n'est pourtant pas acceptable.