

Quick 2 – 18 novembre 2022 – 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.

Dans tout ce devoir, on manipulera les arbres binaires avec les fonctions usuelles : $ArbreVide()$, $Noeud(g, x, d)$, $EstVide(a)$, $FilsGauche(a)$, $FilsDroit(a)$, $Cle(a)$.

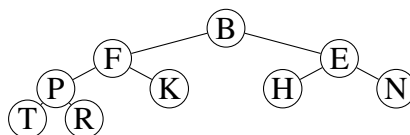
On notera systématiquement :

- n le nombre de nœuds dans l'arbre ;
- et h sa hauteur.

1 Nœuds plus petits que x dans un tas (9 points)

Dans cet exercice, on considère des tas-*min* (où les nœuds sont plus petits que leurs fils), représentés sous forme d'arbres binaires.

Pour tous les exemples donnés dans l'énoncé, on utilisera le tas :



Question 1 : (2 points)

Écrire un algorithme qui, étant donné un tas binaire T et une valeur x , affiche les étiquettes de **tous les nœuds** dont la valeur est **inférieure à x** .

Votre algorithme devra visiter le plus petit nombre de nœuds possible.

Avec le tas donné en exemple :

- si x vaut F on affichera BE ;
- si x vaut Q on affichera $BFPKEHN$

(pas forcément dans cet ordre).

Question 2 : (3 points)

Déterminer la complexité au pire de votre algorithme :

- d'abord en fonction du nombre n de nœuds dans le tas ;
- puis également en fonction de la hauteur h du tas.

On prendra soin de justifier les réponses proposées.

Question 3 : (2 points)

Proposer un algorithme qui affiche les étiquettes de valeurs **supérieures à x** .

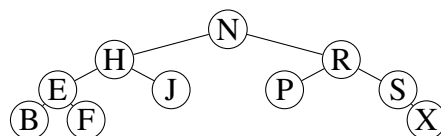
Dans notre exemple, pour $x = M$, on affichera $NPRT$ (pas forcément dans cet ordre).

Question 4 : (2 points)

Pour chacun des deux algorithmes proposés, dites s'il fonctionnerait aussi sur un arbre binaire quelconque et expliquez pourquoi.

2 Premiers éléments dans un arbre binaire de recherche (11 points)

Pour tous les exemples donnés dans l'énoncé, on utilisera l'arbre binaire de recherche :



Il n'est pas demandé de justifier la complexité des algorithmes écrits pour cet exercice.

Question 5 : (2 points)

Écrire un algorithme qui renvoie le plus petit élément dans un arbre binaire de recherche (contenant au moins un nœud).

Dans notre exemple, il faut renvoyer B .

Votre algorithme devra avoir une complexité au pire en $\mathcal{O}(h)$ et visiter le plus petit nombre de nœuds possible.

Question 6 : (5 points)

Écrire un algorithme qui renvoie le 2^e plus petit élément dans un arbre binaire de recherche (contenant au moins 2 nœuds).

Dans notre exemple, il faut renvoyer E .

Représentez tous les cas possibles par des schémas et assurez-vous que votre algorithme les traite correctement.

Votre algorithme devra toujours avoir une complexité au pire en $\mathcal{O}(h)$ et visiter le plus petit nombre de nœuds possible.

Question 7 : (4 points)

Écrire un algorithme qui, étant donné un entier k (inférieur ou égal à n), renvoie le k -ième plus petit élément dans un arbre binaire de recherche.

Dans notre exemple, pour $k = 5$, il faut renvoyer J .

Vous pouvez pour cet algorithme utiliser une structure auxiliaire comme un tableau, une file, une pile... L'usage de variables globales est autorisé, par contre évitez à tout prix les `break` et autres instructions similaires.

Un algorithme qui n'utilise pas de variable globale rapportera plus de points.

Votre algorithme devra avoir une complexité au pire en $\mathcal{O}(n)$.

Question 8 :

Cette question est difficile, et sera valorisée en points de bonus uniquement sous condition d'avoir déjà traité sérieusement le reste de l'énoncé.

Écrire un **autre algorithme** qui résout le **même problème** (renvoyer le k -ième plus petit élément dans un arbre binaire de recherche), mais cette fois sans utiliser de structure auxiliaire, ni modifier l'arbre.

Votre algorithme devra avoir une complexité au pire en $\mathcal{O}(h + k)$ et visiter le plus petit nombre de nœuds possible.

Indication : il est recommandé d'ajouter des paramètres supplémentaires à votre procédure et/ou de renvoyer un couple de valeurs bien choisies.