
Quick 2 – 19 novembre 2021 – 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 manipule les arbres binaires avec les fonctions usuelles :
ArbreVide(), *Noeud(g, x, d)*, *EstVide(a)*, *FilsGauche(a)*, *FilsDroit(a)*, *Cle(a)*.

1 Algorithme de Huffman (8 points)

Question 1 :

À l'aide de l'algorithme de Huffman, calculer un arbre de codage optimal pour la phrase à 35 symboles :

SIRE RAMONS UNIS SANS NOUS CENSURER

Attention à ne pas oublier le caractère espace.

Question 2 :

Coder les 10 premiers caractères de cette phrase selon le code obtenu.

Question 3 :

Décoder le début de mot binaire suivant à l'aide de votre code :

011101101011100011110111011000...

(Le texte obtenu n'aura pas forcément de sens, puisqu'il dépend de votre arbre de codage.)

Question 4 :

Écrire un algorithme qui prend pour arguments un arbre de codage et un mot binaire, et détermine si oui ou non ce mot binaire est le code d'un symbole de l'alphabet.

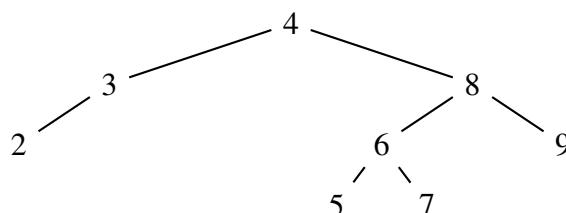
2 Partition d'arbre binaire de recherche (12 points)

On considère un arbre binaire de recherche (ABR) dont les nœuds contiennent des éléments *tous différents*.

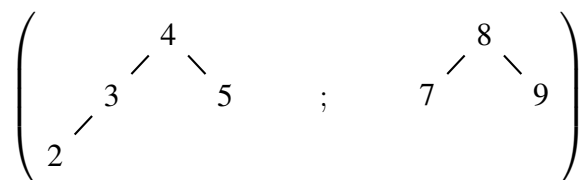
Le but de cet exercice est le suivant : étant donné un ABR A et un élément e **présent dans l'arbre**, créer deux ABR B et C tel que :

- B contient tous les éléments de A inférieurs à e ;
- C contient tous les éléments de A supérieurs à e .

Par exemple, si on partitionne l'arbre suivant par rapport à l'élément 6



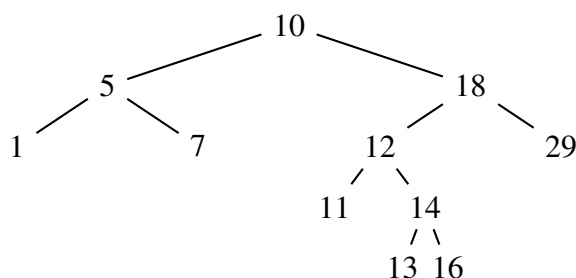
Le résultat pourrait être le couple d'arbres :



La complexité au pire de votre algorithme devra être $\mathcal{O}(h)$, la hauteur de l'arbre. Il est demandé de ne pas utiliser d'autre structure de données ni les fonctions d'insertion ou de suppression dans un ABR.

Question 5 :

Illustrer la construction de ce couple d'arbres sur l'arbre suivant :



- lorsqu'on partitionne par rapport à 10 ;
- lorsqu'on partitionne par rapport à 18 ;
- et enfin lorsqu'on partitionne par rapport à 14.

Question 6 :

Écrire un algorithme *récuratif* qui prend pour argument un arbre binaire de recherche A et un élément e présent dans A , et qui renvoie un *couple* d'ABR contenant les éléments respectivement inférieurs et supérieurs à e .

Il est recommandé d'illustrer le fonctionnement de votre algorithme à l'aide de schémas.