Université
Caen Normandie

Année 2018-2019

Portail Mathématiques, Informatique
Licence deuxième année (L2 Info, L2 maths, L2 MIASHS)

Algorithmique et structures de données
TD 8 – arbres binaires de recherche

Question 1. Rappelez la définition d'un ABR. Dans toute la suite, le coût des procédures sera le nombre de comparaisons entre deux entiers.
Donnez la procédure *rechercheABR(A :ABR,x :entier) :ABR* qui retourne un pointeur sur le nœud contenant la valeur x et None si la valeur n'est pas présente.
Soit n le nombre de nœuds de A et h sa hauteur.
Donnez la complexité dans le pire des cas de *rechercheABR* en fonction de h.
Rappelez l'encadrement de h par n. en déduire la complexité dans le pire des cas en fonction de n lorsque A est un arbre complet ?

Question 2. Donnez la procédure *insereFeuille(A :ABR,x :entier) :ABR* qui retourne un ABR enrichi d'un nœud contenant la valeur x inséré au niveau des feuilles.
Comparer la complexité de cette procédure avec celle de la procédure *rechercheABR*.

Question 3. Construisez tous les ABR possibles en insérant des nœuds étiquetés avec les valeurs de l'ensemble $E_3 = \{1, 2, 3\}$. Pour cela, vous utiliserez l'insertion aux feuilles en insérant les valeurs de E_3 dans tous les ordres possibles. Que remarquez-vous ?

Question 4. Écrivez une procédure *afficheCroissant(A :ABR)* qui affiche toutes les valeurs de A par ordre croissant. Comment modifier la procédure pour obtenir l'ordre décroissant ?

Insertion des nœuds à la racine

Dans beaucoup d'applications, les valeurs les plus recherchées sont celles qui ont été insérées récemment. Dans ce cas, il n'est pas judicieux d'insérer les nouveaux nœuds au niveau des feuilles car le coût de la recherche dépend de la profondeur du nœud. Il est préférable d'ajouter le nœud à la racine.

Pour insérer un nœud de valeur x à la racine de A, on sépare A en deux ABR *inf* et *sup* selon la valeur de coupure x de telle sorte que

$$\begin{array}{rcl} inf & = & \{y \in A : y < x\} \\ sup & = & \{y \in A : y > x\} \end{array}$$

où $x \in A$ signifie que x est la valeur d'un nœud de A.

Nous souhaitons effectuer cette coupure sans créer de nouveau nœud.

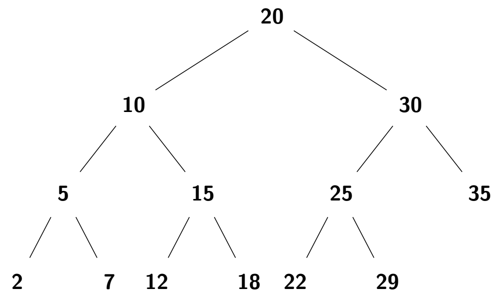
Question 5. Que donne la coupure selon la valeur 20, puis les valeurs 32 et 13 sur l'ABR de la Figure 1.

Question 6. * Écrivez une procédure récursive *coupeSelon(A :ABR, x :entier) :(ABR,ABR)* qui retourne les deux arbres *inf* et *sup* correspondant à la coupure de A selon la valeur x.

Indication : vous considérerez les quatre cas suivants.

1. $A =$ None
2. x est égal à la valeur de la racine de A
3. x est inférieur à la valeur de la racine de A

Figure 1 – Exemple d'ABR



4. x est supérieur à la valeur de la racine de A

Montrez comment appeler récursivement la procédure *coupeSelon* dans les cas 3. et 4.

Question 7. Donnez le coût de la procédure *coupeSelon*. Quelle est la complexité dans le pire des cas en fonction de h et en fonction de n ?

Question 8. Écrivez une procédure *insereALaRacine(A :ABR, x :entier) :ABR* qui retourne un ABR enrichi d'un nœud de valeur x inséré à la racine.

Pour cela, vous utiliserez impérativement la procédure *coupeSelon*.

Question 9. Donnez la complexité dans le pire des cas de la procédure *insereALaRacine* en fonction de h et en fonction de n.

Question 10. * Écrivez une procédure récursive *fusionABR(A :ABR,B :ABR) :ABR* qui retourne un ABR C qui est la fusion des ABR A et B. C doit contenir tous les nœuds de A et B, vous ne devez pas créer de nouveaux nœuds. Cette fusion peut être réalisée de plusieurs façons. Nous procéderons de la manière suivante :

1. si $A = None$ alors on retourne B
2. si $B = None$ alors on retourne A
3. si $A \neq None$ et $B \neq None$, on effectue les instructions suivantes
   — on effectue une coupe de B selon la valeur de la racine de A
   — on obtient deux ABR inf et sup
   — on fusionne inf avec $A \rightarrow gauche$ et sup avec $A \rightarrow droit$.

Question 11. * Donnez une majoration du nombre de comparaisons impliquant chaque valeur de A. En déduire une majoration de la complexité dans le pire des cas en fonction de la hauteur h **et** du nombre de nœuds n de A.

On suppose maintenant que les deux ABR A et B sont des arbres binaires complets de hauteur h et que chaque coupe lors de la fusion produit deux ABR de même taille également complets. Soit $C(h)$ le coût de la fusion de deux ABR complets de hauteur h.

Montrez que l'on a la relation

$$C(h) = h + 2\,C(h-1).$$

On montre que l'on a $C(h) \sim 2^{h+1}$.
En déduire le coût de la procédure avec les ABR complets en fonction de n.
Comparez avec la majoration précédente.