TD 3 : Arbres binaires de recherche

28 Septembre 2017

1 Arbre binaire de recherche symétrique

QUESTION 1 – Écrire la fonction SYMÉTRIQUE qui prend en paramètre un arbre binaire de recherche T et renvoie son symétrique (voir Fig. 1 et Fig. 2). Quelle est sa complexité en fonction de la taille de T ?

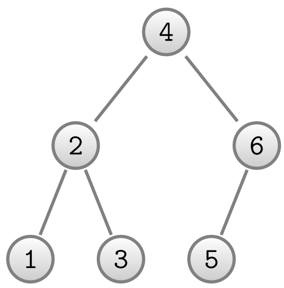

FIGURE 1 – Un arbre binaire T.

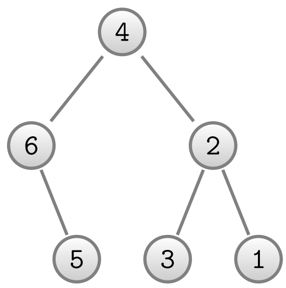

FIGURE 2 – L'arbre symétrique de T.

2 Procédures d'ajout d'un élément

QUESTION 2 – On considère dans un premier temps la procédure d'ajout d'un élément dans un arbre binaire de recherche vue en cours. À quel endroit dans l'arbre l'élément est-il ajouté ?
QUESTION 3 – Proposer une procédure d'ajout d'un élément à la racine d'un arbre binaire de recherche. On pourra pour ce faire utiliser une procédure intermédiaire.
QUESTION 4 – Ajouter successivement 10, 4 et 16 à la racine de l'arbre A de la Figure 3.
QUESTION 5 – Discuter de l'avantage d'une telle procédure d'ajout en comparaison à la procédure d'ajout aux feuilles vue en cours.

3 Extraction d'une plage de valeur

QUESTION 6 – Ecrire une fonction $\text{INTERVALLE}(A, min, max)$ qui prend en entrée un arbre de recherche A et deux entiers min et max, puis renvoie un arbre de recherche qui contient toutes les clés de A appartenant à l'intervalle $[min, max]$.

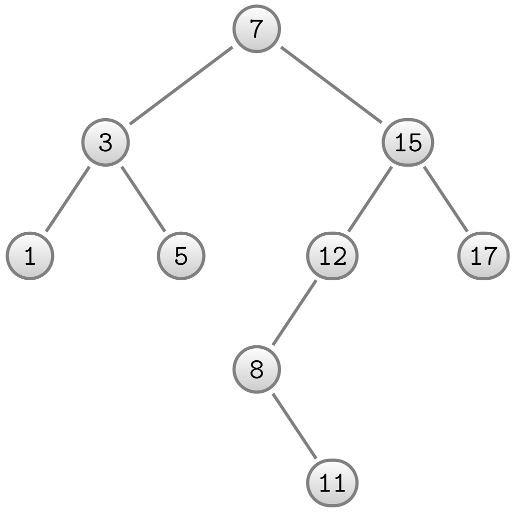

FIGURE 3 – Un arbre binaire de recherche A.

4 Complexité moyenne des algorithmes sur les ABR

Le but de cet exercice est de montrer que les procédures de recherche, d'ajout et de suppression dans un arbre binaire de recherche sont, en moyenne, de complexité logarithmique. Pour cela, nous avons besoin d'un modèle rigoureux de la phrase : « en moyenne un arbre binaire de recherche possédant n éléments distincts est de la forme... ». Ce modèle est fourni par les ABR_n.

La classe ABR_n représente les arbres binaires de recherche qui possèdent n éléments distincts, et est définie récursivement par :
— l'arbre vide est un ABR_0,
— pour tout $n \geq 1$, si $A \in ABR_n$, alors A a été obtenu par ajout aux feuilles d'un élément dans un arbre appartenant à ABR_{n-1}.

QUESTION 7 – Construire ABR_3.

Un ABR_n est donc obtenu par ajouts (aux feuilles) successifs de n éléments distincts $a_1, \ldots, a_n$. L'ordre d'ajout des a_i détermine la forme de l'ABR_n que l'on considère.

QUESTION 8 – Combien y a-t-il d'ordres d'ajout différents ?

Si T est un ABR, on définit :
— $LCE(T) = \sum_f p(f)$, la somme des profondeurs des feuilles de T,
— $PE(T)$ la profondeur moyenne des feuilles de T,
— PE_n la profondeur moyenne d'ajout dans un arbre ABR_n.

On cherche à obtenir une formule de récurrence liant PE_n à PE_{n-1}. Pour ce faire, on définit le complété T_c d'un arbre T comme l'arbre T auquel on ajoute des feuilles carrées à chaque emplacement où un ajout au feuille est possible (*cf.* Figure 4 et Figure 5).

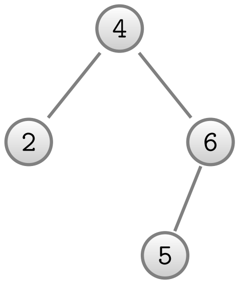

FIGURE 4 – Un arbre binaire T.

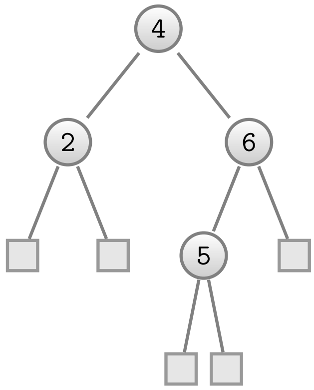

FIGURE 5 – T_c le complété de l'arbre T.

QUESTION 9 – Soit T un ABR_n. Trouver une formule liant $PE(T_c)$ et $LCE(T_c)$.
QUESTION 10 – Soit T un ABR_n. Il est obtenu par ajout aux feuilles d'un élément dans un arbre de ABR_{n-1}, noté T'. Trouver une formule liant $LCE(T_c)$ et $LCE(T'_c)$.
QUESTION 11 – Prouver que :

$$PE_n = \frac{1}{(n+1)!} \sum_{T \in ABR_n} LCE(T_c)$$

QUESTION 12 – Prouver que

$$PE_n - PE_{n-1} = \frac{2}{n+1}$$

QUESTION 13 – Conclure.