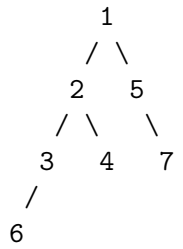
TD 5

Exercice 1 : Arbre binaire complet

La longueur du chemin intérieur d'un arbre binaire complet est la somme, restreinte à tous les noeuds internes de l'arbre, de la profondeur de chaque noeud. De même, la longueur du chemin extérieur est la somme, restreinte à toutes les feuilles de l'arbre, de la profondeur de chaque feuille. On considère un arbre binaire complet avec n noeuds internes, un chemin intérieur de longueur i, et un chemin extérieur de longueur e. Démontrer que $e = i + 2n$ (commencer par le vérifier sur 2 ou 3 exemples que vous construirez).

Exercice 2 : Affichage d'arbres

On supposera que la mise en oeuvre des arbres binaires est faite par pointeurs "filsGauche, filsDroit, père".
- Ecrire une fonction récursive qui étant donné un arbre binaire B, imprime l'étiquette de chaque noeud de l'arbre, selon un parcours préfixé.
- Ecrire une fonction non récursive qui étant donné un arbre binaire B, imprime l'étiquette de chaque noeud de l'arbre, selon un parcours préfixé. Si besoin, vous pouvez utiliser une pile comme structure de données auxiliaire.
- Appliquez pas à pas vos deux fonctions sur l'arbre suivant :



Exercice 3 : Arbre binaire de recherche

Dessiner des arbres binaires de recherche de hauteur 2, 3, 4, 5 et 6 pour le même ensemble d'entiers l, 4, 5, 10, 16, 17, 21.

Exercice 4 : Séquences de noeuds parcourus

Nous supposons que les entiers compris entre 1 et 1000 sont disposés dans un arbre binaire de recherche, et on souhaite retrouver le nombre 363. Parmi les séquences suivantes, lesquelles ne pourraient pas être la séquence de noeuds parcourus ?

a) 2,252,401,398,330,344,397,363.
b) 924,220,911,244,898,258,362,363.
c) 925,202,911,240,912,245,363.
d) 2,399,387,219,266,382,381,278,363.
e) 935,278,347,621,299,392,358,363.

Exercice 5 : Contre-exemple

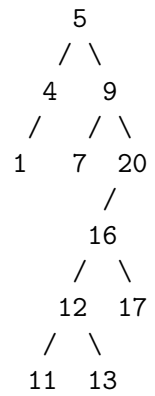
Le professeur Bunyan pense avoir découvert une remarquable propriété des arbres binaires de recherche. On suppose que la recherche d'un élément k dans un arbre binaire de recherche se termine sur une feuille. On considère trois ensembles : A, les clés situées à gauche du chemin de recherche ; B, celles situées le long du chemin de recherche ; et C, les clés situées à droite du chemin de recherche. Le professeur Bunyan affirme que trois $a \in A, b \in B$ et $c \in C$ quelconques doivent satisfaire $a \leq b \leq c$. Donner un contre-exemple aussi petit que possible.

Exercice 6 : Successeur

a) Donnez l'arbre binaire de recherche, noté ABR dans toute la suite, obtenu en ajoutant, à l'aide de l'algorithme d'ajout aux feuilles, successivement les éléments 15, 5, 20, 23, 3, 18, 9, 6, 7, 13,11.
b) Que donne le parcours infixé de l'ABR précédent ?
c) Ecrire l'algorithme, que l'on appellera min-ABR, qui prend en argument un ABR et retourne son plus petit élément. Quelle est sa complexité ? L'appliquer à l'ABR obtenu en a).
d) Etant donné un élément x dans un ABR, expliquez et justifiez en le démontrant comment l'on trouve l'élément qui lui est immédiatement supérieur dans l'ABR. Cet élément est en fait le successeur de x si l'on fait un parcours infixé de l'ABR
Donnez la fonction, que l'on appellera suc-ABR, qui prend en argument le noeud x et retourne son successeur.
Appliquez votre fonction sur l'arbre de la question a) pour obtenir : le successeur de l'élément 15, puis le successeur de l'élément 13, enfin le successeur de l'élément 3.
e) Expliquez comment l'on peut simuler un parcours infixé dans un ABR à l'aide des fonctions min-ABR et suc-ABR.

Exercice 7 : Suppression

Soit l'arbre binaire de recherche ci-dessous.



a) Donner les nouveaux ABR obtenus en supprimant successivement les éléments 13, puis 20, puis 5.
b) Ecrire la fonction supprimemax(A) qui prend en entrée l'arbre A en retourne le plus grand élément de A tout en le supprimant de A.
c) Ecrire la fonction Supprimer(x, A) qui supprime l'élément x dans l'ABR A. Vous pourrez vous aider, si besoin, de la fonction précédente.