

ALGORITHMIQUE AVANCEE
TD 6 ING-4-GLSI

Arbre Binaire et Arbre Binaire de Recherche

Exercice 1 :

1. Ecrire une fonction récursive **Taille** permettant de trouver la taille d'un arbre binaire donné. La taille d'un arbre est le nombre de nœuds de cet arbre.
2. Ecrire une fonction **Feuille** qui indique si le nœud racine est une feuille.
3. Ecrire une fonction récursive **NbFeuilles** permettant de calculer le nombre de feuilles d'un arbre binaire donné.
4. Ecrire une fonction récursive **Hauteur** permettant de trouver la hauteur d'un arbre binaire donné. La hauteur d'un arbre est le nombre de nœuds entre la racine et la feuille de la plus longue branche partant de la racine.
5. Ecrire une fonction récursive **Dégénéré** qui indique si un arbre binaire est dégénéré. Un arbre binaire est dégénéré si pour chaque nœud interne, il n'y a qu'un successeur, le dernier nœud étant une feuille.
6. Ecrire une procédure récursive **Exchange** qui échange les fils gauche et droit de chaque nœud d'un arbre binaire donné.

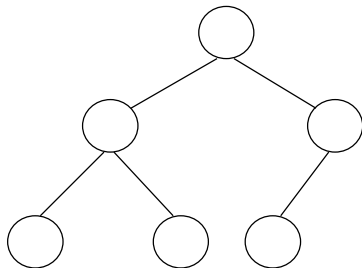


FIG. 1 – Avant Exchange

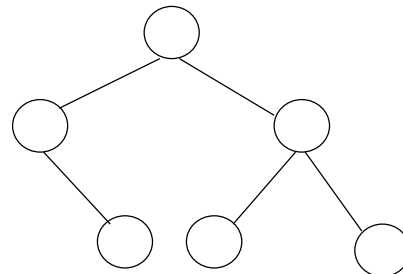


FIG. 2 – Après Exchange

Exercice 2 :

1. Ecrire une fonction récursive **Tester** permettant de tester si un arbre binaire est un arbre binaire de recherche.
2. Ecrire une procédure itérative **Insérer** permettant d'insérer un élément donné dans un arbre binaire de recherche.
3. Ecrire une procédure récursive **Construire** permettant la construction d'un arbre binaire de recherche à partir d'une liste chaînée d'entiers.
4. Ecrire une fonction récursive **RechercheOcc** permettant de chercher la $n^{\text{ème}}$ occurrence d'une valeur x dans un arbre binaire de recherche, la fonction retourne l'adresse du nœud ayant pour valeur x et qui apparaît à la $n^{\text{ème}}$ position dans l'arbre binaire de recherche.

Exercice 3 :

1. Ecrire une proc **Affiche_Croissant** permettant d'afficher les valeurs des nœuds d'un ABR de manière croissante.
2. Ecrire une proc **Affiche_arbre** permettant d'afficher les valeurs des nœuds d'un arbre ABR de manière à voir la structure de l'arbre. Un nœud sera affiché par (g,v,d) où g est le sous-arbre gauche, v la valeur du nœud et d le sous-arbre droit. Par exemple, l'arbre présenté dans la figure ci-dessous sera affiché par : (((_,1,_) , 3,_) , 4, ((_,6,_) , 6, ((_,7,_) , 9,_))). Les '_' indiquent les sous-arbres vides.

