

Travaux Dirigés d'informatique
Algorithmique
Les arbres binaires.

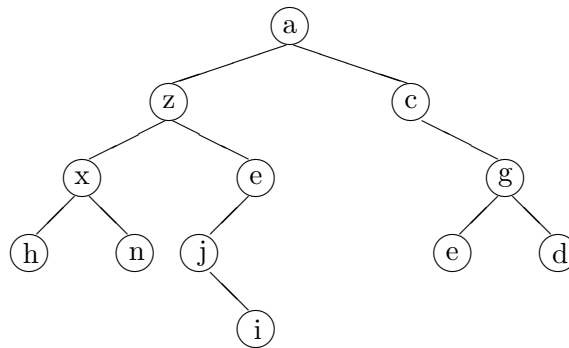


Figure 1: Un arbre binaire

1. Vocabulaire

(a) Donner les étiquettes de :

- la racine,
- des feuilles,
- des nœuds de hauteur 2,
- des nœuds de niveau 3,
- des nœuds sur le chemin $a \rightarrow j$.

(b) Donner le nombre de noeuds, le nombre de noeuds internes, le nombre de feuilles.

(c) Donner les étiquettes des nœuds visités lors d'un parcours en profondeur avec traitement:

- préfixe (racine, fils gauche, fils droit),
- suffixe (fils gauche, fils droit, racine),
- infixé (fils gauche, racine, fils droit).

(d) Donner les étiquettes des nœuds visités lors d'un parcours en largeur.

(e) Hauteur

On rappelle que la hauteur d'un arbre est le maximum de liens entre la racine et une feuille. Donner la hauteur d'un arbre réduit à une seule feuille, d'un arbre vide et de l'arbre ci-dessus. Quelle est la hauteur du nœud d'étiquette 'z' ?

2. Progammmation

On utilise les types C:

```
typedef struct noeud{
    int valeur;
    struct noeud *fg;
    struct noeud *fd;
}Noeud,*Arbre;
```

- (a) Écrire une fonction renvoyant la hauteur d'un arbre.
- (b) Écrire une fonction renvoyant le nombre de noeuds de l'arbre.
- (c) Écrire une fonction renvoyant le nombre de noeuds internes de l'arbre.
- (d) Écrire une fonction renvoyant le nombre de feuilles de l'arbre.
- (e) Écrire une fonction renvoyant le nombre de noeuds de l'arbre possédant exactement 2 fils.

3. Parcours

Écrire une fonction affichant les étiquettes des noeuds d'un arbre lors d'un parcours en profondeur,

- ordre préfixe,
- ordre infixe,
- ordre suffixe.

Rappelez la définition d'une file (ainsi que les opérations possibles) et donnez l'algorithme permettant d'afficher un arbre binaire par un parcours en largeur.

4. Description d'un arbre étiqueté par des entiers strictement positifs

Dans cette méthode, les nœuds sont décrits par leur étiquette lors par un parcours en profondeur préfixe, un fils vide est symbolisé par 0. Construisez les deux arbres binaires décrits par les deux séquences suivantes:

- 8 3 2 1 0 0 5 0 0 7 0 0 9 4 0 0 6 0 0
- 4 3 4 5 2 0 0 0 1 3 0 0 0 0 3 5 0 0 2 1 0 0 6 0 0

- 5. Écrire la fonction `void afficheArbPositif(Arbre a)` qui affiche la suite des entiers décrivant un arbre
- 6. Décrire la méthode pour construire un arbre binaire à partir d'une suite d'entiers positifs (On pourra supposer la suite correcte).
- 7. Écrire la fonction `int construitArbPositif(Arbre *a)` qui construit un arbre binaire à partir d'une suite d'entiers valide (saisie au clavier).
- 8. **Arbre héréditairement gauche**

Un arbre strictement binaire (chaque noeud a 0 ou 2 fils) est dit héréditairement gauche s'il est réduit à sa racine, ou s'il vérifie les trois conditions suivantes:

- le sous-arbre gauche a un nombre de feuilles supérieur ou égal à celui de droite,
 - le sous-arbre gauche est héréditairement gauche,
 - le sous-arbre droit est héréditairement gauche.
- (a) Tracer tous les sous-arbres héréditairement gauche à trois feuilles, puis à cinq feuilles.
 - (b) Écrire une fonction `int estHG(Arbre A)` qui retourne 1 si l'arbre passé en argument (supposé strictement binaire) est héréditairement gauche et 0 sinon.