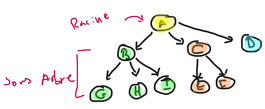
EL MAJJODI ABDELJALIL
Cours Les Arbres avec les algorithmes en C
2022

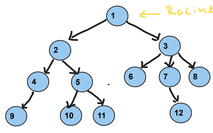
→ les arbres sont des structures non linéaires.
→ noeud (A) → Racine → sous Arbres



fils (A) : B - C - D
feuille : G, H, I, E, F, D
noeud I : B, C

→ les fils d'un noeud sont les racines de ses Arbres.
→ feuille : noeud sans fils.
→ noeud Interne : noeud qui n'est pas feuille.

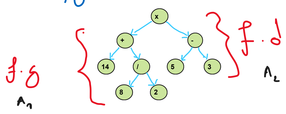
Vocabulaire :



* Degré d'un noeud : le nombre de fils de ce noeud.
→ D(n) = (2, 3)
* Degré d'un Arbre : plus grand degré des noeud de l'arbre.
→ D(A) = D(3) = 3 ⇝ 3-aire.
* Taille d'un noeud : nombre Total des noeuds :
→ T(A) = 12
* Hauteur (profondeur, niveau) : longueur de chemin qui relie la racine à ce noeud.
→ H(5) = 2
* Hauteur d'un Arbre : le profondeur maximum de ses noeuds.
→ H(A) = 3

Arbre Binaire

→ Arbre où chaque noeud a au plus deux fils.
→ fils droit / gauche.

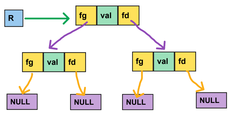


→ écrit sous forme suivante :

A = < x, A1, A2 >

→ Représentation :
1) noeud :

2) Arbre est désigné par un ptr sur sa Racine.

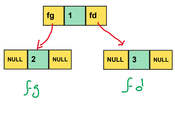


→ Arbre vide est représenté par le pointeur NULL.

→ Code :C :

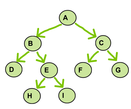
```c
typedef struct node{
    int val;
    struct node *fg;
    struct node *fd;
}Node;
```


```c
int main()
{
    Node *r,*fg,*fd;

    r=(Node*)malloc(sizeof(Node));
    fg=(Node*)malloc(sizeof(Node));
    fd=(Node*)malloc(sizeof(Node));

    r->val=1;
    fg->val=2;
    fd->val=3;

    r->fg=fg;     r->fd=fd;
    fg->fg=NULL;  fg->fd=NULL;
    fd->fg=NULL;  fd->fd=NULL;


    return 0;
}
```



→ Parcour d'un Arbre :
* On distingue deux catégories de parcours :
→ Parcours en profondeur.
→ Parcours en hauteur.

1) Parcours en profondeur :
a) Parcours préfixé : (R, G, D) ⇝ (Racine, fg, fd)



⇒ A B D E H I C F G

⇝ Code En C :

```c
void rgd(Node *R){
    if(R){
        printf("%d ",r->val);
        rgd(r->fg);
        rgd(r->fd);
    }
}
```