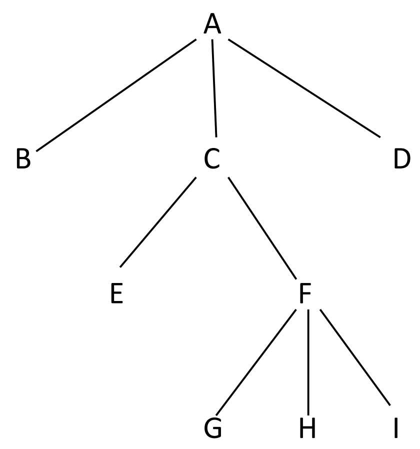
LES ARBRES

Définitions (Selon NIST)

NIST: **N**ational **I**nstitute of **S**tandards and **T**echnology.

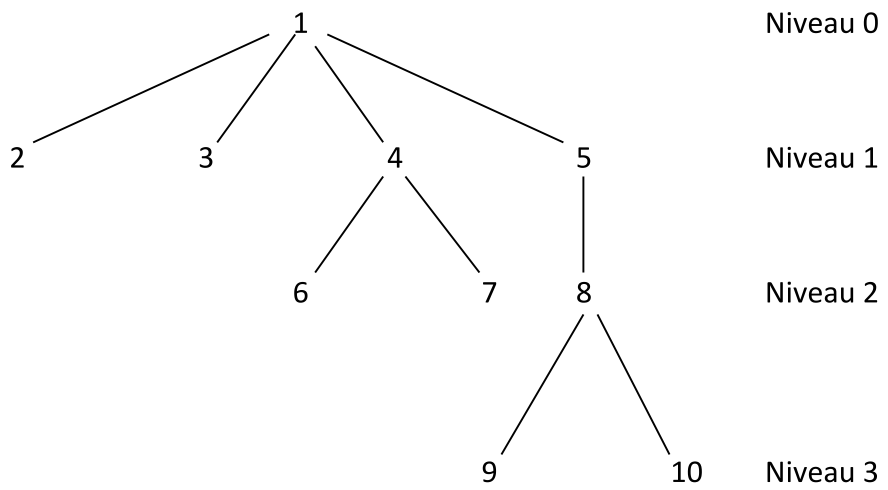
- Un arbre est une structure de données qui peut se présentée sous forme d'une hiérarchie dont chaque élément est appelé nœud ; le nœud initial étant appelé racine.
  Chaque nœud peut avoir des nœuds descendants directs qu'on appelle fils, chaque nœud fils a un père et les nœuds qui n'ont pas de fils sont appelés feuilles.
  Exemple :



  Racine : A
  Feuilles : B, E, G, H, I et D
  Nœuds intermédiaires : C et F

- Cet arbre est un arbre n-aire, c'est-à-dire il a au moins un nœud qui a plus de deux fils : A et F.
- Un arbre est dit binaire, si tous ses nœuds ont au plus deux fils : fils gauche et fils droit.

Soit l'arbre **A** suivant :



- **<u>Hauteur, profondeur ou niveau d'un nœud X.</u>**

  H(X) = Nombre de nœuds à partir de la racine jusqu'au nœud X.

  Exemple : H(9) = 4 car pour aller de la racine 1 jusqu'au nœud X , on doit passer par 1-5 puis 5-8 et enfin 8-9 donc 4 nœuds qui sont 1, 5, 8 et 9.

  Par conséquent H(racine) = 1.

  Remarque : D'autres auteurs ne comptent pas la racine, dans ce cas H(9) = 3 !

- **<u>Chemin d'un nœud X.</u>**

  C'est de donner la liste des nœuds de la racine jusqu'au nœud X.

  Exemple : C(10) = (1, 5, 8, 10), C(7) = (1, 4, 7).

  Donc on peut dire que H(X)= nombre de nœuds de C(X).

- **<u>Degré d'un nœud X.</u>**

  C'est le nombre de ses descendants directs c'est-à-dire le nombre de ses fils.

  Exemple : d°(1) = 4 car 1 a 4 fils. d°(8) = 2. d°(7)= 0.

  <u>Remarque :</u> Si tous les nœuds d'un arbre sont de degré 1 : On le nomme arbre dégénéré et que c'est en fait une liste !

- **<u>Hauteur ou profondeur d'un arbre A.</u>**

  H(A) = Nombre de nœuds du chemin le plus long.

  Pour notre exemple H(A) = 4.

- **<u>Degré d'un arbre A.</u>**

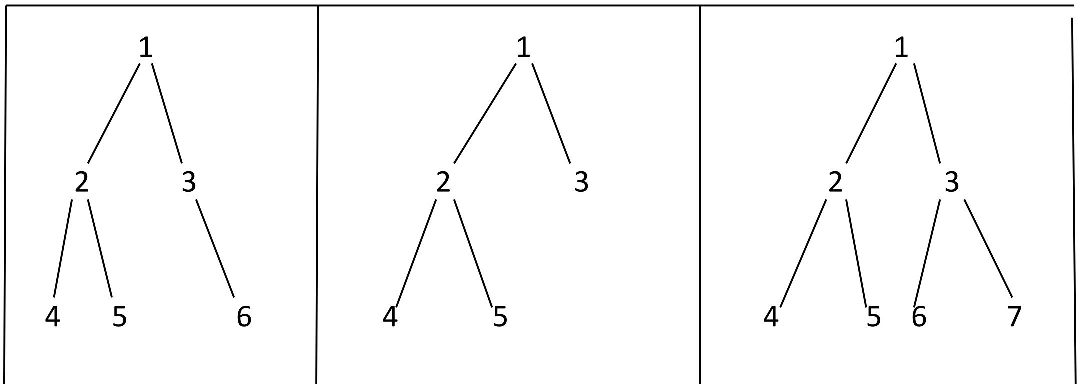
  d°(A) = maximum de tous les degrés des nœuds de l'arbre.

  Comme d°(1) = 4 et c'est le maximum donc d°(A) = 4.

- **<u>Taille d'un arbre A.</u>**
  C'est le nombre de nœuds qui le composent.
  Donc T(A) = 10.

- Les arbres binaires sont les plus utilisés c'est-à-dire les arbres de degré 2.
- Un arbre binaire entier est un arbre dont tous ses nœuds possèdent zéro ou deux fils.
- Un arbre binaire parfait est un arbre binaire entier dans lequel toutes ses feuilles sont à la même distance de la racine.

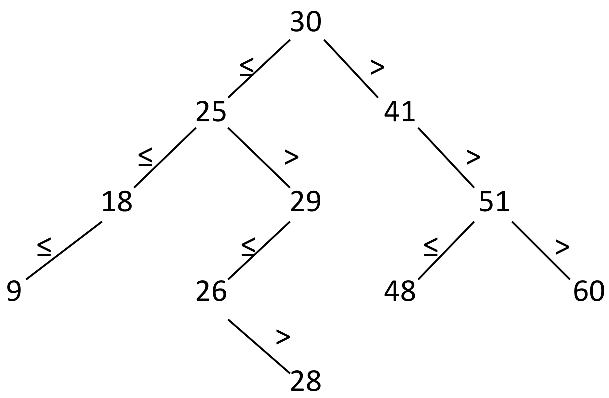
<u>Illustrons cela sur un petit schéma :</u>



Arbre binaire | **Arbre binaire entier** | **Arbre binaire parfait**

<u>NB :</u> L'arbre binaire parfait est parfois appelé arbre binaire complet.

- **<u>Arbre de recherche</u>** est un arbre binaire tel que pour chaque nœud la valeur (info) de son fils gauche lui est inférieure ou égale et la valeur de son fils droit lui est strictement supérieure.
  Exemple :



Cet arbre est dit arbre de recherche.

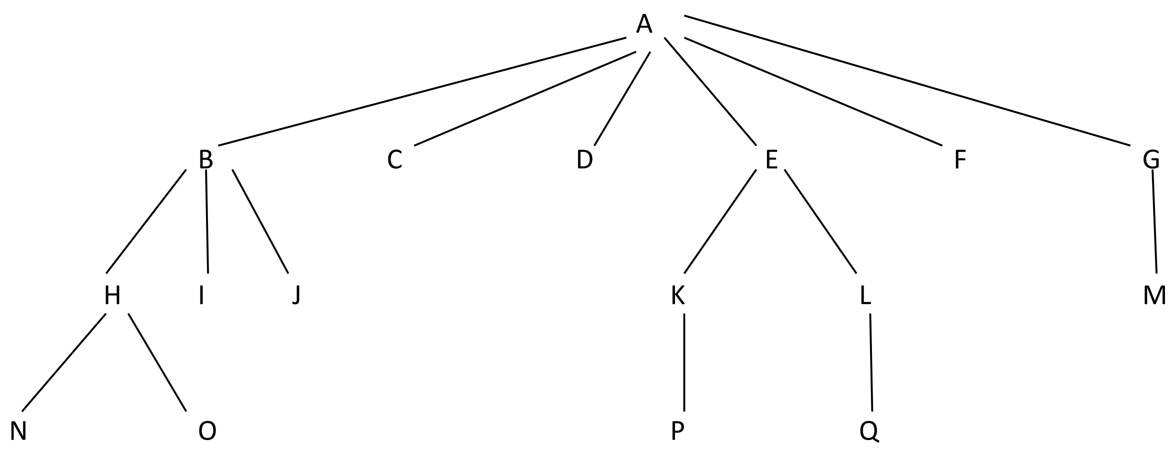
NB : Dans un arbre de recherche l'information qui se trouve dans chaque nœud est appelée clef (KEY).

- Dans un arbre qu'il soit n-aire ou binaire, on peut parler de sous arbres.
  - La racine et ses descendants forment l'arbre.
  - N'importe quel autre nœud et ses descendants forment un sous arbre.

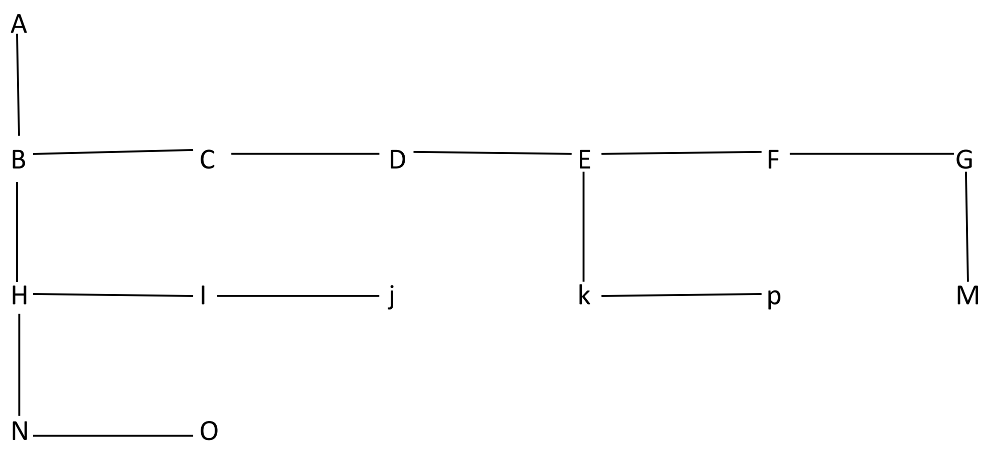
TRANSFORMATION D'UN ARBRE N-AIRE EN UN ARBRE BINAIRE

- Il existe une **injection**, entre les arbres n-aires et les arbres binaires, qui est spécialement utilisée par le langage « LISP » pour représenter les arbres n-aires en arbres binaires.
- Pratiquement, c'est une méthode très simple qui se résume en:
    - Ecrire la racine.
    - Ecrire à la verticale le fils le plus à gauche.
    - Ecrire à l'horizontale de tous les frères de ce fils.
    - Faire la même chose pour chaque nœud de l'arbre.
    - Une fois le schéma de l'arbre est dessiné, faire une quart de tour vers la gauche et on aura l'arbre binaire correspondant.

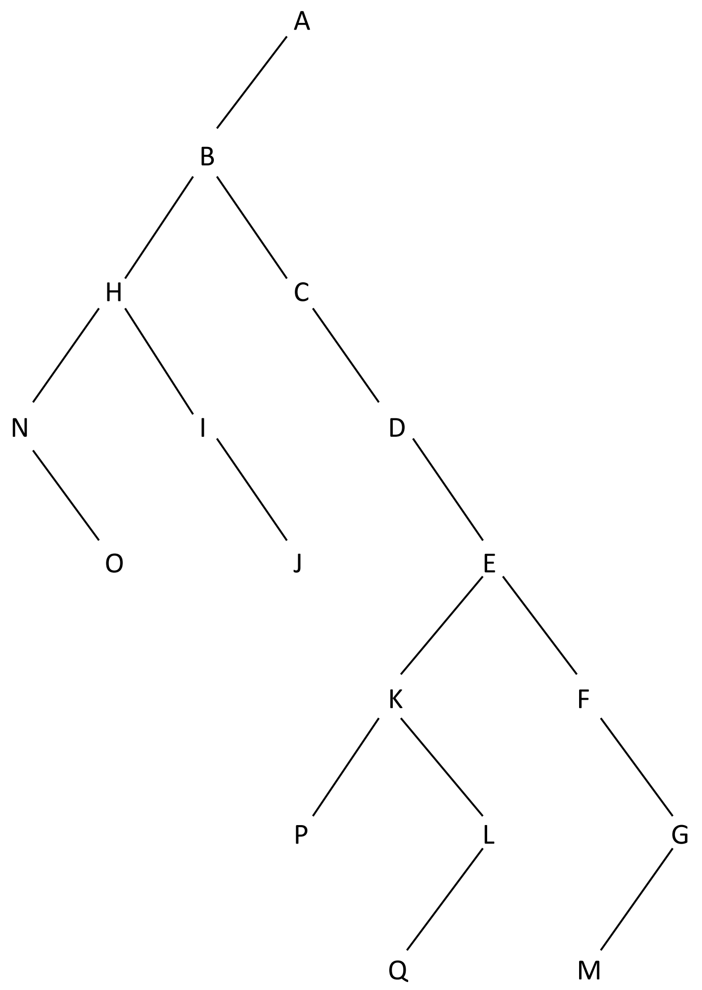
- **Exemple:** Soit l'arbre suivant.



Première étape: Le schéma.



Deuxième étape: Le quart de tour du schéma.



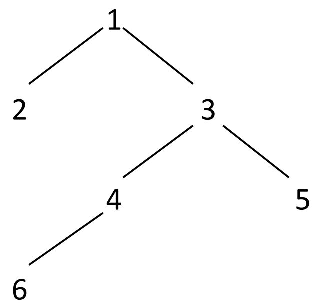
REPRESENTATION GRAPHIQUE (SCHEMATIQUE) D'UN ARBRE BINAIRE

- On peut le représenter sous forme d'un tableau mais son traitement n'est pas aisé, c'est pour cela qu'on le représente sous forme d'une structure chaînée où chaque nœud est composé de trois champs : L'un pour la valeur du nœud, l'autre pour l'adresse du fils gauche et le troisième pour le fils droit.

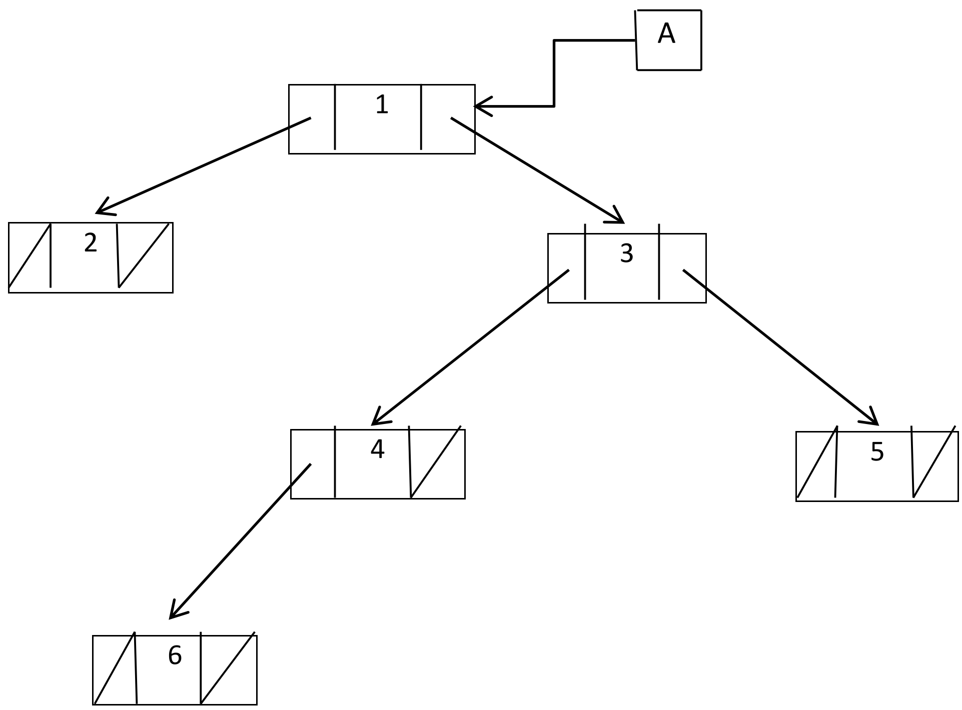
Node

Gauche	Info	Droit

<u>Soit l'arbre suivant :</u>



<u>Il sera représenté par :</u>



Si on appelle ceci par l'arbre **A** donc il est représenté par l'adresse de sa racine.

PARCOURS D'UN ARBRE BINAIRE

Nous allons étudier trois (03) parcours : Préfixe, infixe et postfixe qui sont aussi appelés préordre, inordre et postordre.

Ecrivons des procédures récursives en pseudo-code (Algorithmique) pour réaliser ce travail.

- **Parcours préfixe :**

```
Procédure Visiter_prefixe( A : arbre) ;
Début
        Ecrire (A->info) ;
        Si (A->gauche) ≠ NULL  alors
            Visiter_prefixe(A->gauche) ;
        Fin si
        Si (A->droit) ≠ NULL  alors
            Visiter_prefixe(A->droit) ;
        Fin si
Fin
```

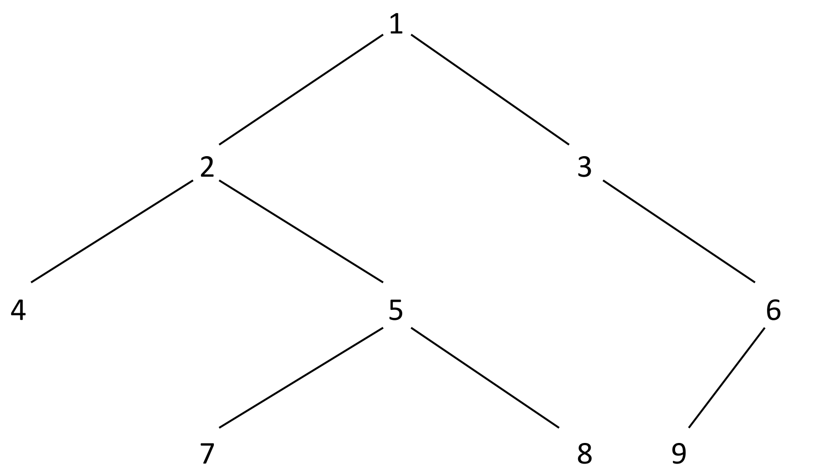
- **Parcours postfixe**

```
Procédure Visiter_postfixe( A : arbre) ;
Début
        Si (A->gauche) ≠ NULL  alors
            Visiter_postfixe(A->gauche) ;
        Fin si
        Si (A->droit) ≠ NULL  alors
            Visiter_postfixe(A->droit) ;
        Fin si
        Ecrire (A->info) ;
Fin
```

- **Parcours infixe**

```
Procédure Visiter_infixe( A : arbre) ;
Début
        Si (A->gauche) ≠ NULL  alors
            Visiter_infixe(A->gauche) ;
        Fin si
        Ecrire (A->info) ;
        Si (A->droit) ≠ NULL  alors
            Visiter_infixe(A->droit) ;
        Fin si
Fin
```

- On essaie de faire tourner les trois procédures sur l'arbre suivant :



  - Le parcours infixe donne :
    4, 2, 7, 5, 8, 1, 3, 9 et 6

  - Le parcours postfixe donne :
    4, 7, 8, 5, 2, 9, 6, 3 et 1

  - Le parcours préfixe donne :

    1, 2, 4, 5, 7, 8, 3, 6 et 9

Une petite remarque :

- Les langages fonctionnels comme **CAML, Haskell**, … sont des langages qui utilisent la structure de données « arbre » avec aisance !
- Pour les langages orientés objet, exemple **java**, la structure « arbre » est implémentée en tant qu'une classe.