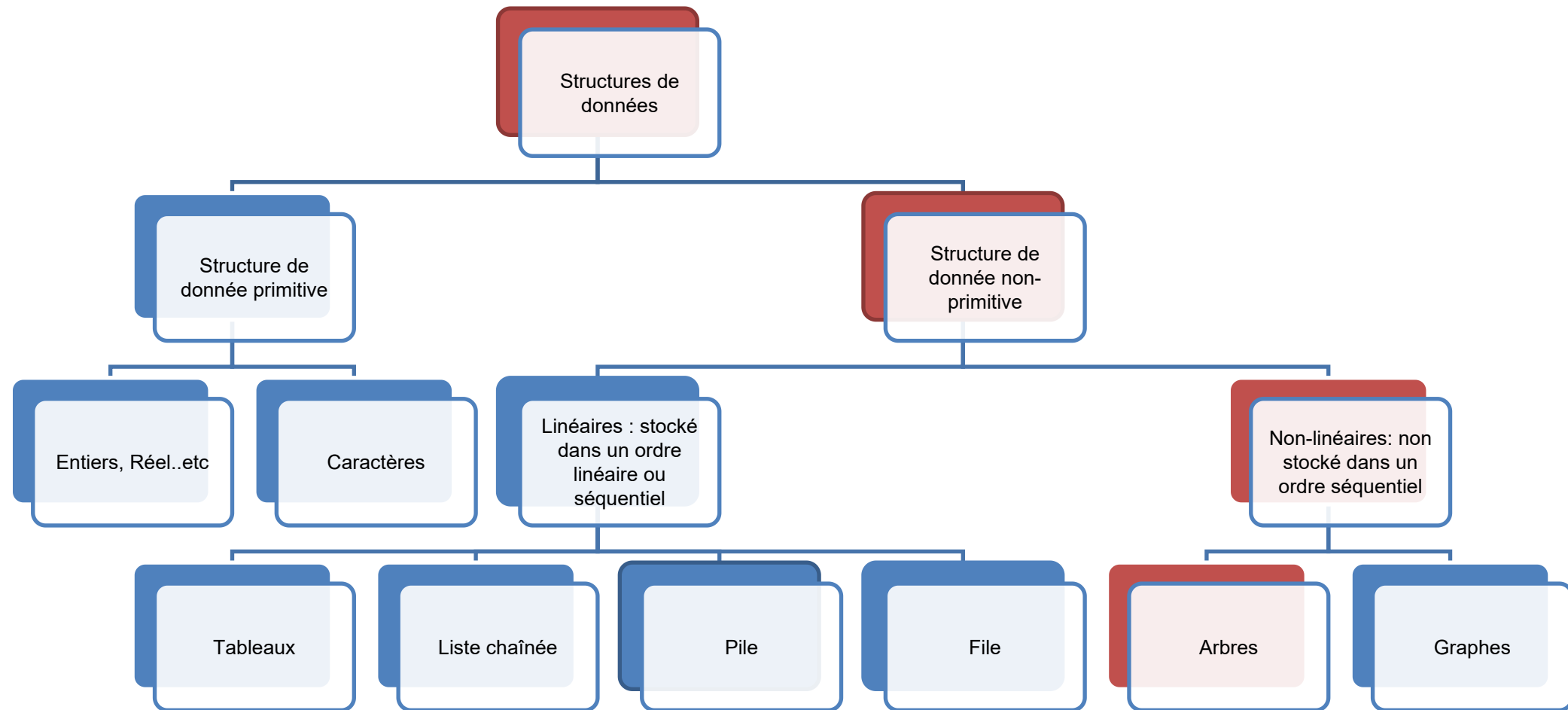


Programmation & Algorithmique II

CM 12 : Les arbres

CLASSIFICATION DES STRUCTURES DE DONNÉES

➤ Structures de données primitives et non primitives



Préambule

- Un arbre est une structure qui est principalement utilisée pour stocker des données de nature hiérarchique.

- arbres généraux

PLAN

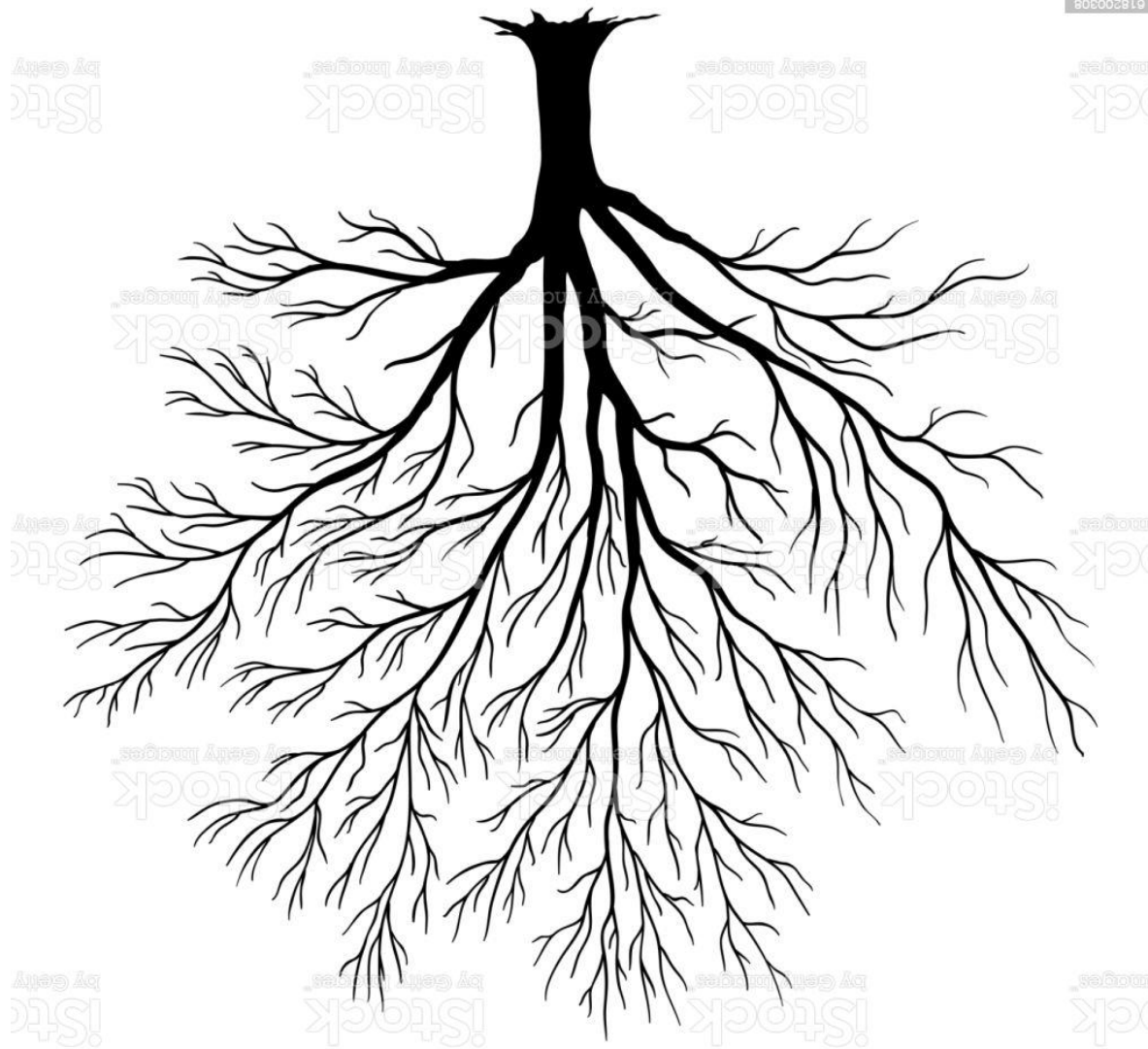
1. Introduction
2. Définition
3. Types des arbres
 1. Arbres généraux
 2. Forêts
 3. Arbres binaires
 1. Arbres binaires complets
 1. Représentation en mémoire
 2. Représentation chaînée des arbres binaires
 3. Représentation contiguë des arbres binaires
 4. Créer un arbre binaire à partir d'un arbre général
 5. Parcours d'un arbre binaire
 6. Arbres de recherche binaire (plus tard)

Introduction (1)



618200308

Introduction (2)



APPLICATIONS

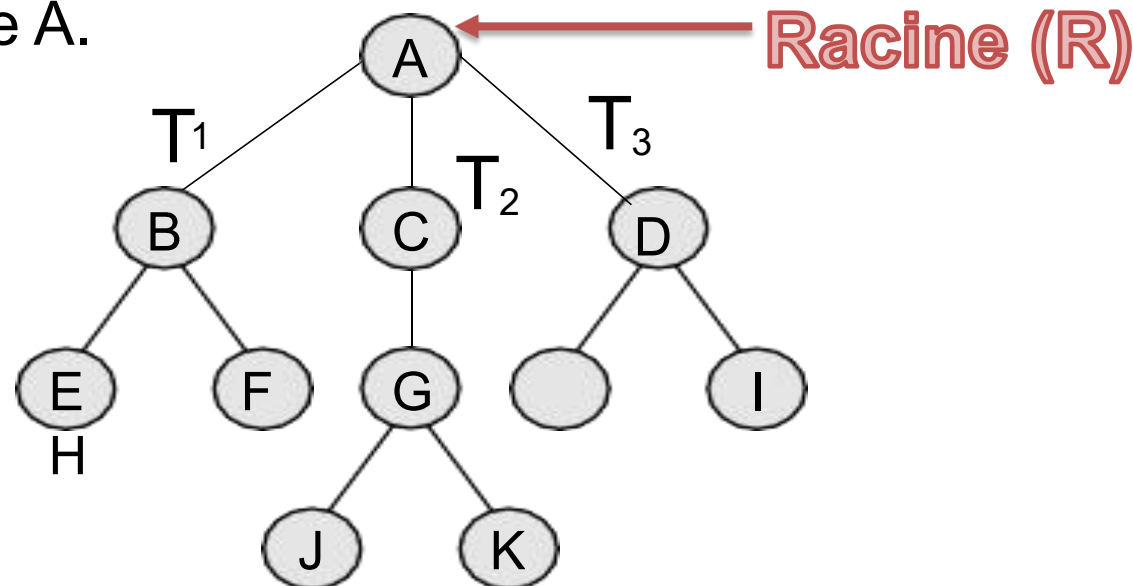
De nombreuses applications s'appuient sur l'utilisation des arbres, on peut citer:

- Stocker des données simples et complexes.
- Implémenter les tables de hachage, les ensembles et les cartes.
- Indexer un grand nombre d'enregistrements (les arbres B) dans les bases de données.
- Compilateurs.
- Répertoires de système de fichiers

DÉFINITION

➤ QU'EST-CE QU'UN ARBRE (Tree en anglais)?

- Un arbre est défini de façon récursive comme un ensemble d'un ou plusieurs nœuds
- Un nœud est désigné comme la racine de l'arbre et tous les nœuds restants peuvent être divisés en ensembles non vides dont chacun est un sous-arbre de la racine.
- La figure suivante montre un arbre où le nœud A est le nœud racine; les nœuds B, C et D sont des enfants du nœud racine et forment des sous-arbres T₁, T₂ et T₃ de l'arbre avec la nœud racine A.



DÉFINITION

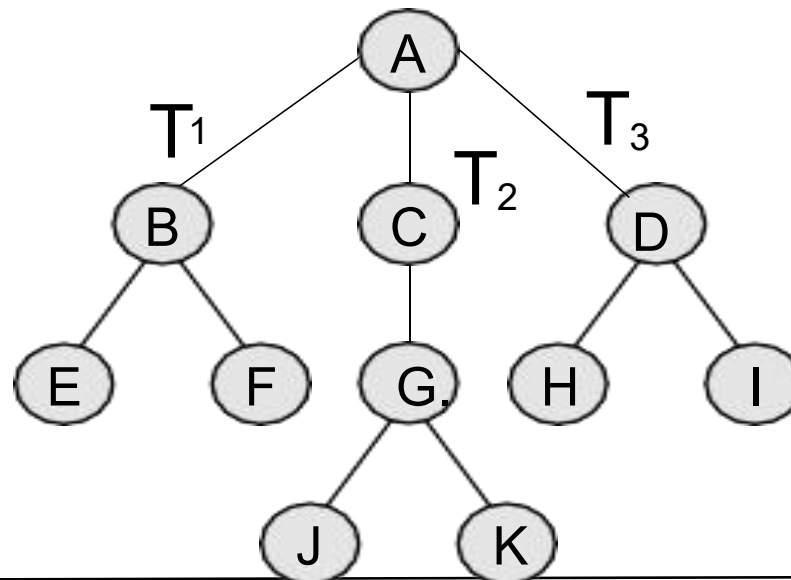
• **Nœud racine** Le nœud racine **R** est le nœud le plus haut de l'arbre. Si **R** = NULL, cela signifie que l'arbre est vide.

Sous-arbres Si le nœud racine **R** n'est pas NULL, alors les arbres T1, T2, et T3 sont appelés les sous-arbres de R.

Nœud feuille Un nœud qui n'a pas d'enfants est appelé le nœud feuille ou le nœud terminal.

Chemin Une séquence de arcs consécutifs est appelée chemin. Par exemple, le chemin du nœud racine A au nœud I est A, D, et I.

Nœud ancêtre Un ancêtre d'un nœud est n'importe quel nœud prédécesseur sur le chemin de la racine à ce nœud. Le nœud racine n'a pas d'ancêtres. Dans l'arbre ci-dessous, les nœuds A, C et G sont les ancêtres du nœud K.

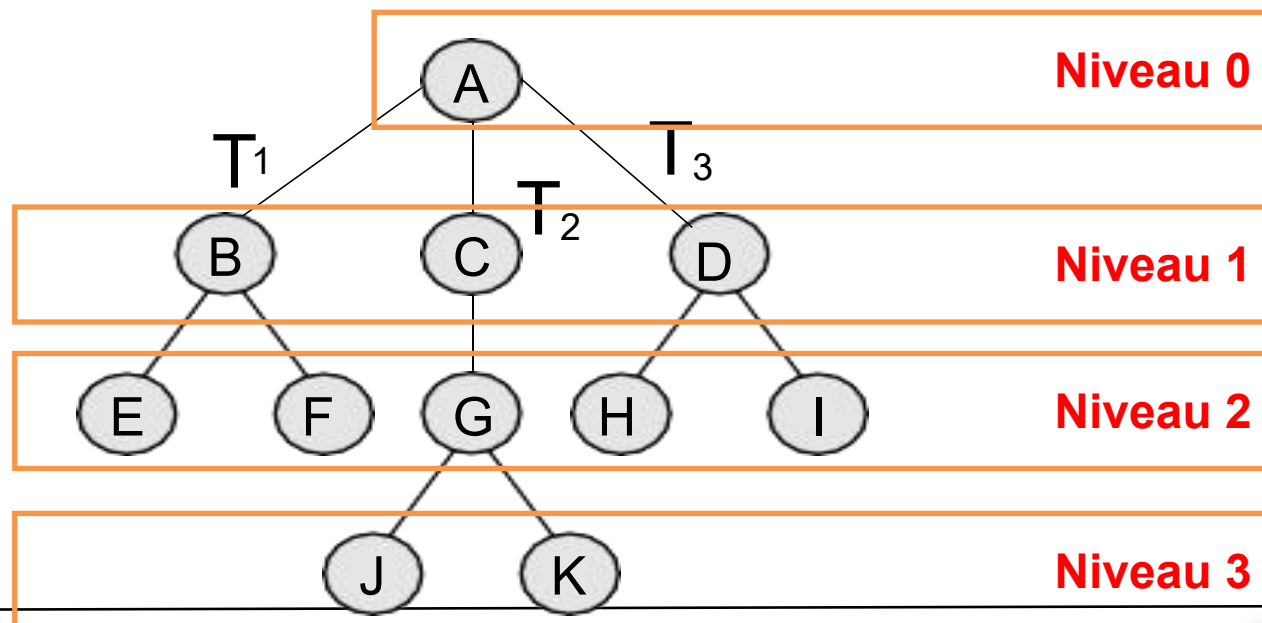


DÉFINITION

• **Nœud descendant** Un nœud descendant est n'importe quel nœud successeur sur n'importe quel chemin du nœud à un nœud de feuille. Les nœuds de feuilles n'ont pas de descendants. Dans l'arbre dessous, les nœuds C, G, J et K sont les descendants du nœud A.

Niveau Chaque nœud dans l'arbre se voit attribuer un niveau de telle sorte que le nœud racine est au niveau 0, les enfants du nœud racine sont au niveau numéro 1. Ainsi, chaque nœud est à un niveau supérieur à son parent. Par conséquent, tous les nœuds d'enfant ont un numéro de niveau donné par le numéro de niveau du parent + 1.

Degré Le degré d'un nœud est égal au nombre d'enfants qu'un nœud possède. Le degré d'un nœud feuille est nul.



LES TYPES D'ARBRES

Il existe plusieurs types d'arbre:

- Arbres généraux
- Forêts
- Arbres binaires
- Arbres binaires de recherche
- Arbres d'expression
- Arbres de tournoi

ARBRES GÉNÉRAUX

Les arbres généraux sont des structures de données qui stockent les éléments hiérarchiquement. Le nœud supérieur d'un arbre est le nœud racine et chaque nœud, à l'exception de la racine, a un parent. Un nœud dans un arbre général (à l'exception des nœuds feuilles) peut avoir zéro sous-arbre ou plus.

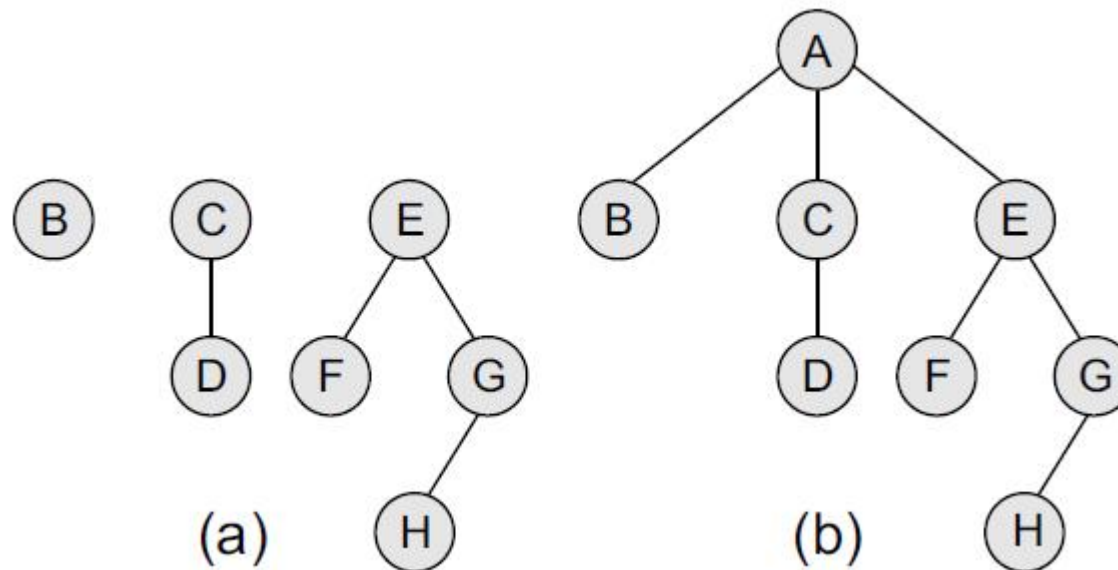
Les arbres généraux qui ont 3 sous-arbres par nœud sont appelés arbres ternaires. Toutefois, le nombre de sous-arbres pour n'importe quel nœud peut être variable. Par exemple, un nœud peut avoir 1 sous-arbre, tandis qu'un autre nœud peut avoir 3 sous-arbres.

Un arbre général est souvent transformé en arbre binaire afin de simplifier les opérations de recherche, de traversée, d'ajout et de suppression des nœuds,

FORÊTS

Une forêt est une union disjointe d'arbres. Un ensemble d'arbres disjoints (ou forêts) est obtenu en supprimant la racine et les bords reliant le nœud racine aux nœuds au niveau 1. Par exemple l'arbre (b) devient la forêt (a).

Nous pouvons convertir une forêt en un arbre en additionnant un seul nœud comme le nœud racine de l'arbre.



ARBRES BINAIRES

Un arbre binaire est une structure de données qui est définie comme une collection d'éléments appelés nœuds. Dans un arbre binaire l'élément supérieur est appelé le nœud racine et **chaque nœud a 0, 1, ou au plus 2 enfants**.

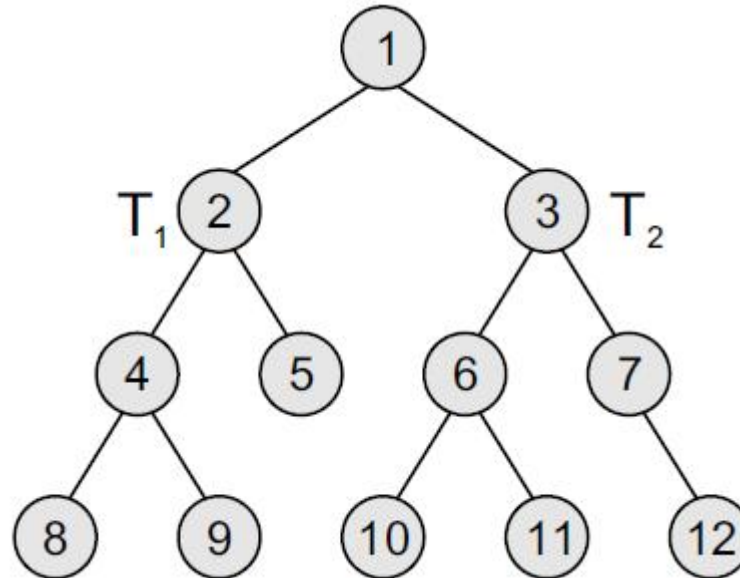
Un nœud qui n'a pas d'enfants est appelé un nœud feuille ou nœud terminal.

Chaque nœud contient un élément de **données**, un pointeur à gauche qui pointe le nœud **fils gauche** et un pointeur à droite qui pointe vers le nœud **fils droit**. Enfin un pointeur **racine** pointe vers le nœud racine. si **racine** = NULL, alors cela signifie que l'arbre est vide.

ARBRES BINAIRES

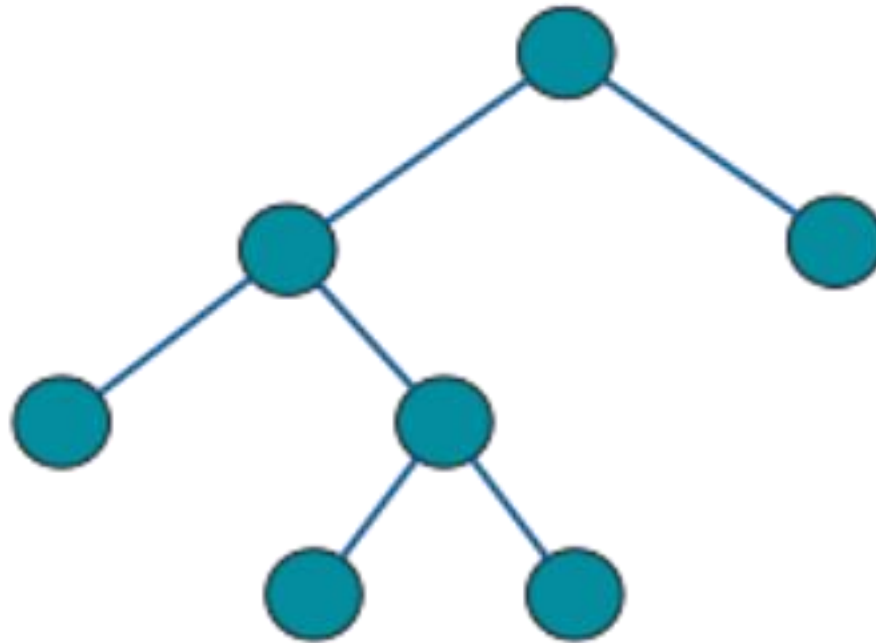
Dans la figure, **R** est le nœud racine et les deux arbres **T1** et **T2** sont appelés sous-arbres gauche et droit de **R**.

Notez que le sous-arbre gauche du nœud racine se compose des nœuds: 2, 4, 5, 8 et 9. De même, le sous-arbre droit du nœud racine se compose de nœuds : 3, 6, 7, 10, 11 et 12.



ARBRE BINAIRE STRICT (FULL BINARY TREE)

- Un **arbre binaire strict** est un arbre dont tous les nœuds possèdent zéro ou deux fils.

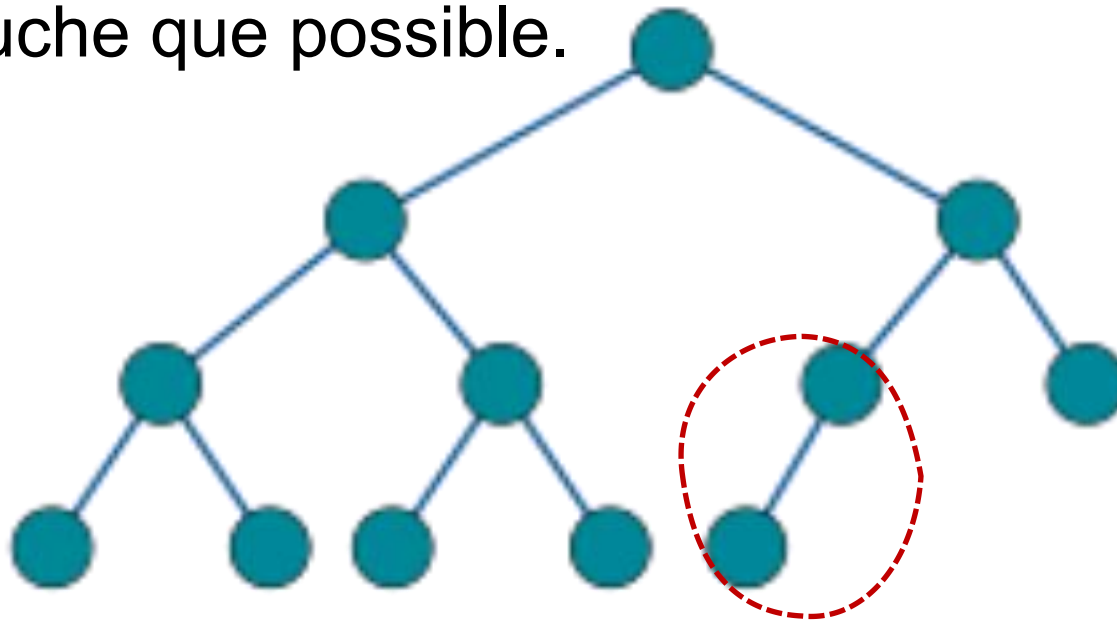


Un arbre strict

ARBRES BINAIRES COMPLETS

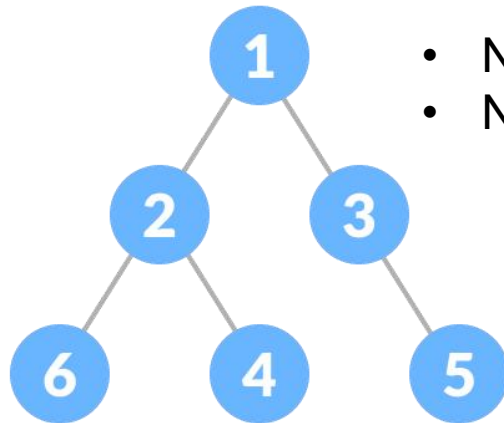
Un arbre binaire complet est un arbre binaire qui satisfait deux propriétés:

- 1- Tout d'abord, dans un arbre binaire complet, chaque niveau, sauf peut-être le dernier, est complètement rempli.
- 2- Deuxièmement, tous les nœuds apparaissent aussi loin à gauche que possible.

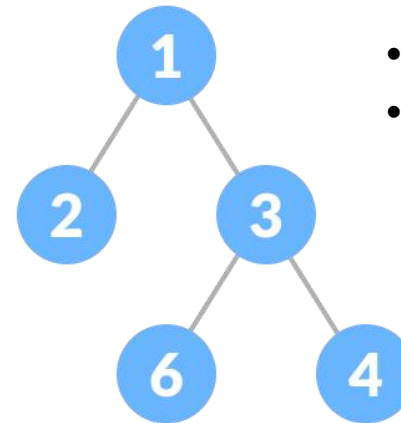


Un arbre complet mais pas strict

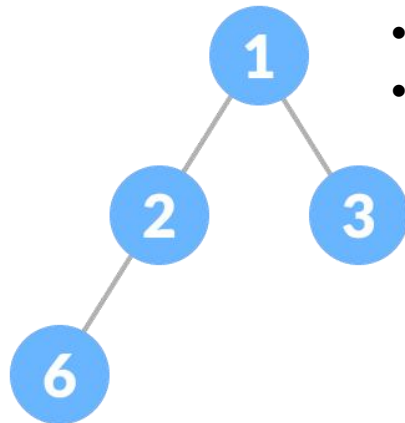
EXERCICE: COMPLET OU STRICT?



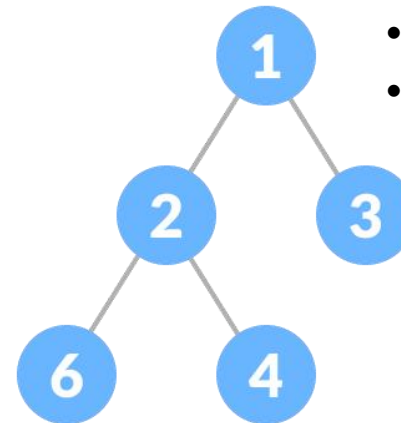
- Non complet
- Non strict



- Non complet
- strict



- complet
- Non strict



- complet
- strict

REPRÉSENTATION DES ARBRES BINAIRES DANS LA MÉMOIRE

Dans la mémoire de l'ordinateur, un arbre binaire peut être maintenu soit en utilisant

- une représentation chaînée
- une représentation contiguë (par tableau).

REPRÉSENTATION CHAÎNÉE DES ARBRES BINAIRES

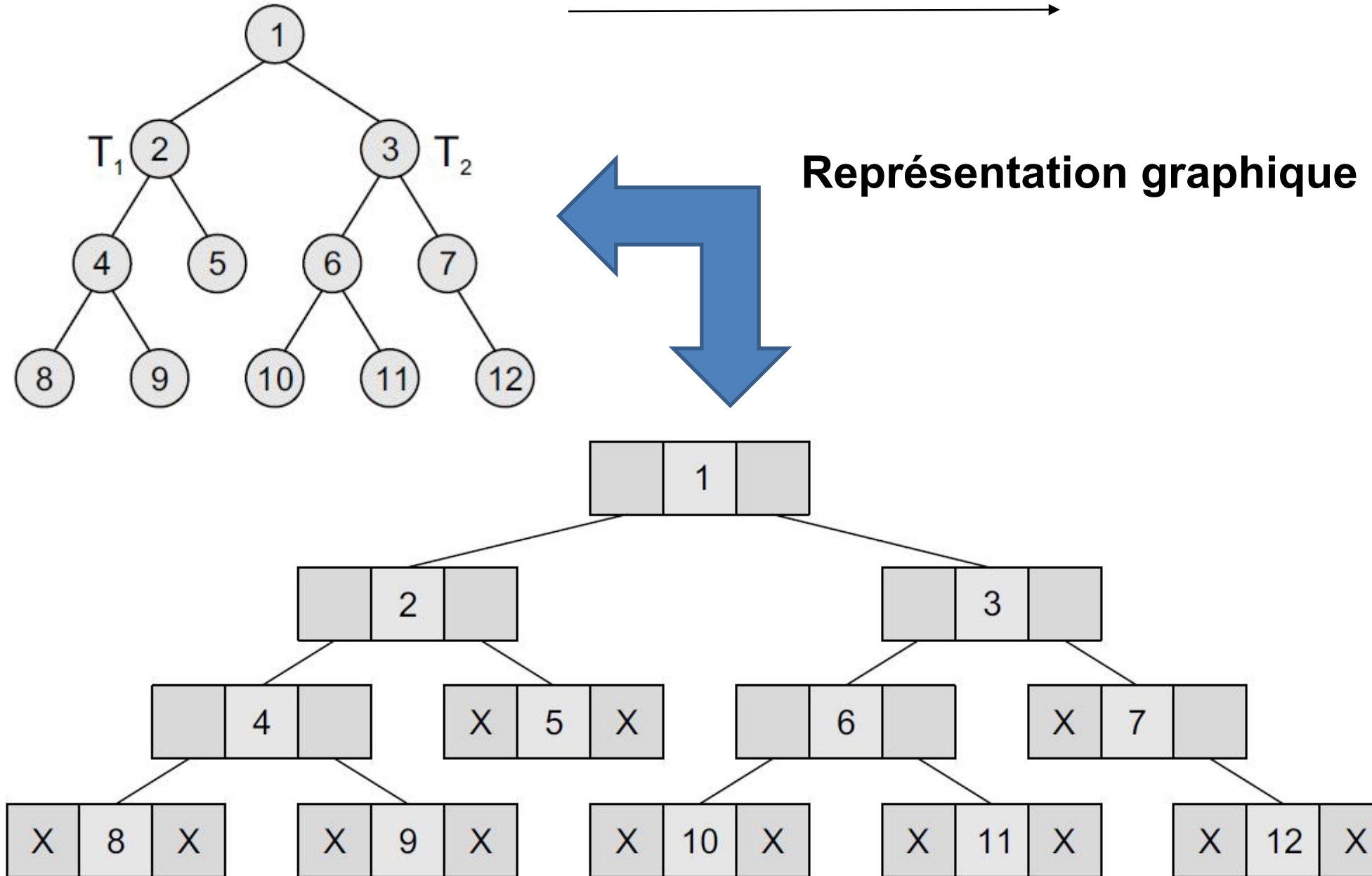
Dans la représentation chaînée d'un arbre binaire, chaque nœud aura trois parties : l'élément de données, un pointeur vers le nœud gauche et un pointeur vers le nœud droit. Ainsi, en C, l'arbre binaire est construit avec un type de nœud donné ci-dessous.

```
struct node {  
    struct node *left;  
    int data;  
    struct node *right;  
};
```

Chaque arbre binaire a un pointeur ROOT, qui pointe vers le nœud racine (élément le plus haut) de l'arbre.

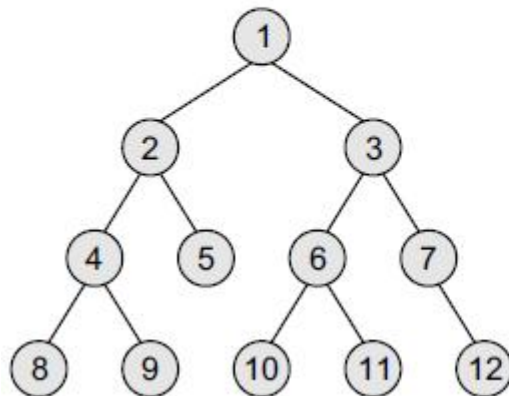
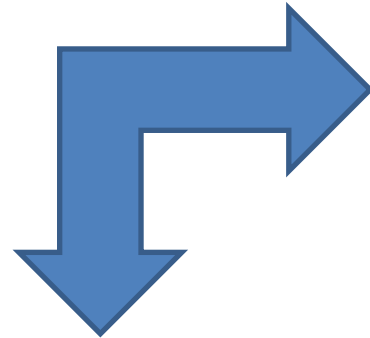
Si ROOT = NULL, alors l'arbre est vide.

REPRÉSENTATION CHAÎNÉE DES ARBRES BINAIRES



REPRÉSENTATION CHAÎNÉE DES ARBRES BINAIRES

Représentation en mémoire



	LEFT	DATA	RIGHT
1	-1	8	-1
2	-1	10	-1
3	5	1	8
4			
5	9	2	14
6			
7			
8	20	3	11
9	1	4	12
10			
11	-1	7	18
12	-1	9	-1
13			
14	-1	5	-1
15			
16	-1	11	-1
17			
18	-1	12	-1
19			
20	2	6	16

ROOT
3

15
AVAIL

REPRÉSENTATION CONTIGUË DES ARBRES BINAIRES

Dans la représentation contiguë de arbres est fait utilisant un tableau unidimensionnelle.

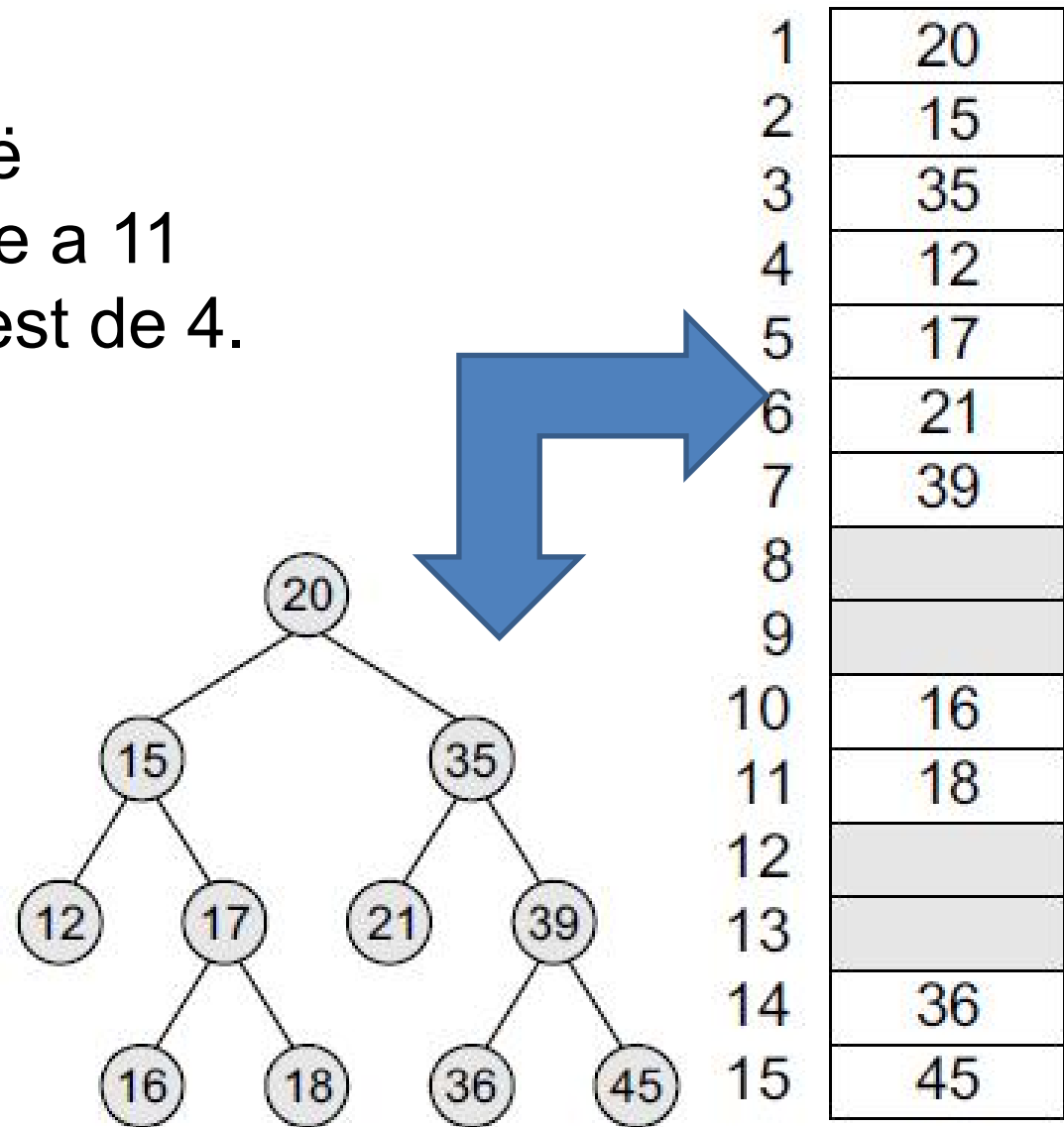
Bien qu'il s'agit de la technique la plus simple pour la représentation de la mémoire, elle est inefficace car elle nécessite beaucoup d'espace mémoire.

Cette représentation suit les règles suivantes:

- Un tableau unidimensionnel, appelé TREE, est utilisé pour stocker les éléments de l'arbre.
- La racine (ROOT) de l'arbre sera stockée à la première case du tableau.
- Les enfants d'un nœud stocké dans la case K seront stockés aux cas $(2 * K)$ et $(2 * K + 1)$.
- La taille maximale de l'ARBRE de tableau est donnée comme $(2^h - 1)$, où h est l'hauteur de l'arbre.
- L'arbre ou sous-arbre vide est spécifié à l'aide de NULL.

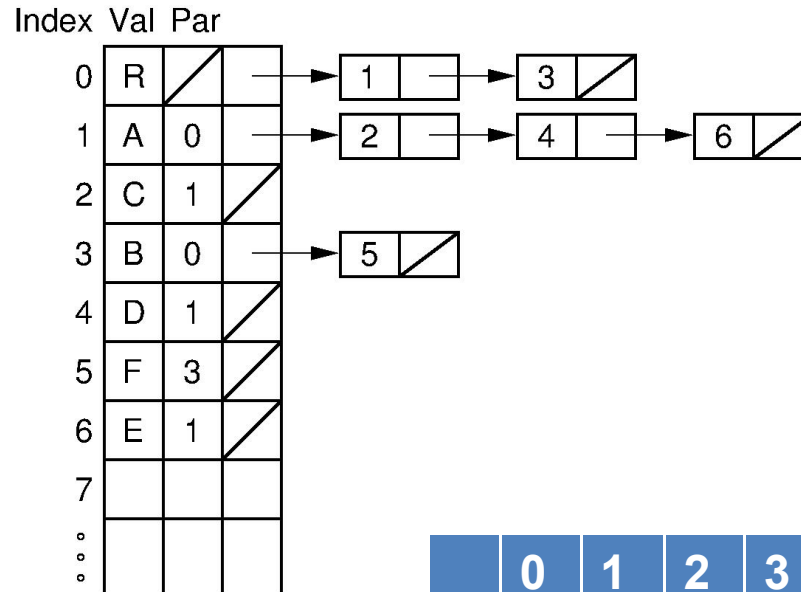
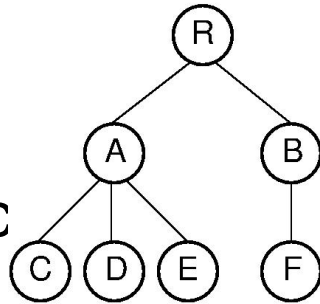
REPRÉSENTATION CONTIGUË DES ARBRES BINAIRES

Un arbre binaire et sa représentation contiguë correspondante. L'arbre a 11 nœuds, et sa hauteur est de 4.



Représentation des arbres généraux

- Un type de
 - Par liste c



- Par m

index	val
0	R
1	A
2	C
3	B
4	D
5	F
6	E

	0	1	2	3	4	5	6
0	0	1	0	1	0	0	0
1	1	0	1	0	1	0	1
2	0	1	0	0	0	0	0
3	1	0	0	0	0	1	0
4	0	1	0	0	0	0	0
5	0	0	0	1	0	0	0
6	0	1	0	0	0	0	0

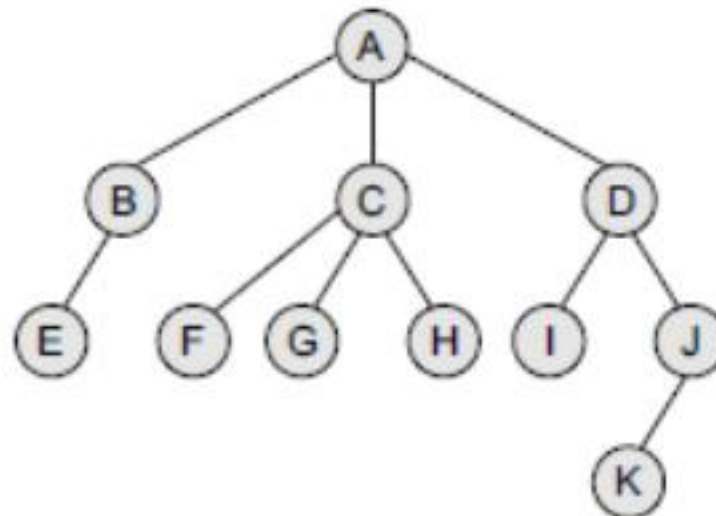
CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Les règles de conversion d'un arbre général en arbre binaire sont données ci-dessous. Notez qu'un arbre général est converti en arbre binaire et non en arbre de recherche binaire.

Règle 1: Racine de l'arbre binaire = Racine de l'arbre général

Règle 2: Fils gauche d'un nœud dans l'arbre binaire = Fils le plus à gauche du nœud dans l'arbre général

Règle 3: Fils droit d'un nœud dans l'arbre binaire = « Frère » droit du nœud dans l'arbre général

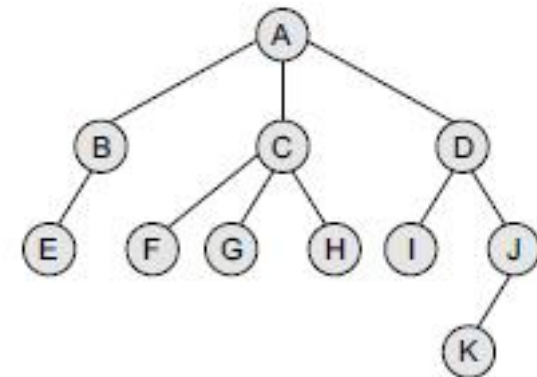
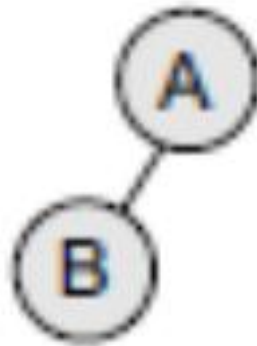


CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 1: Nœud A est la racine de l'arbre général, il sera donc aussi la racine de l'arbre binaire.

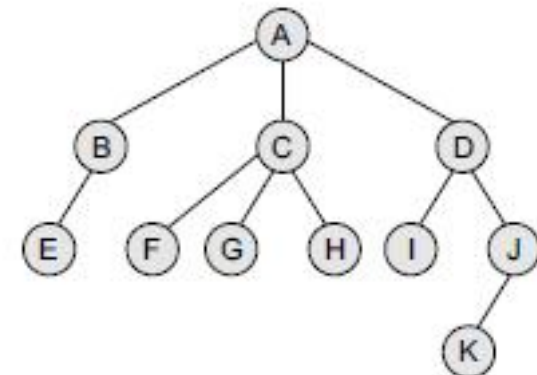
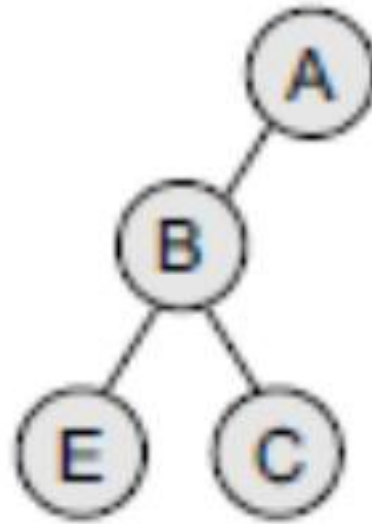


Étape 2: Le nœud fils gauche du nœud A est le nœud fils le plus à gauche du nœud A dans l'arbre général et le fils droit du nœud A est le "frère" droit du nœud A dans l'arbre général. Puisque le nœud A n'a pas de frère droit dans l'arbre général, il n'a pas d'enfant droit dans l'arbre binaire.



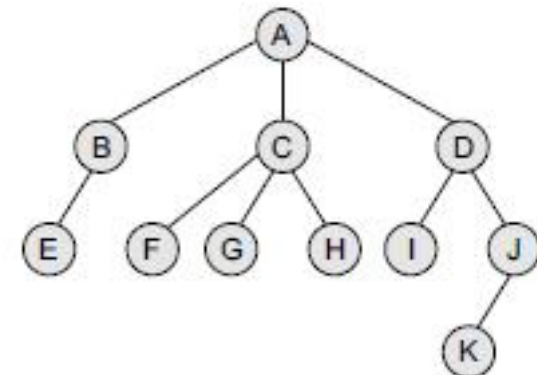
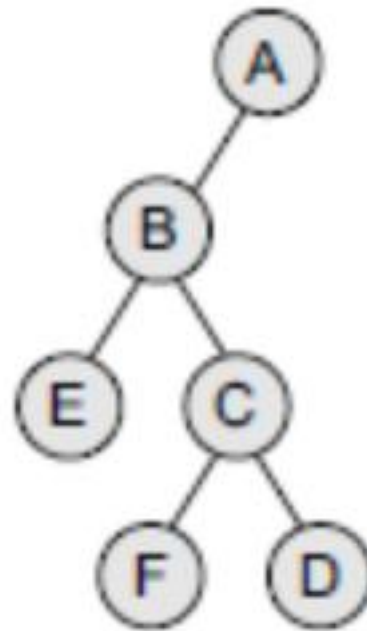
CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 3: Maintenant traitons le nœud B. Le fils gauche de B est E et son fils droit est C (frère droit dans l'arbre général).



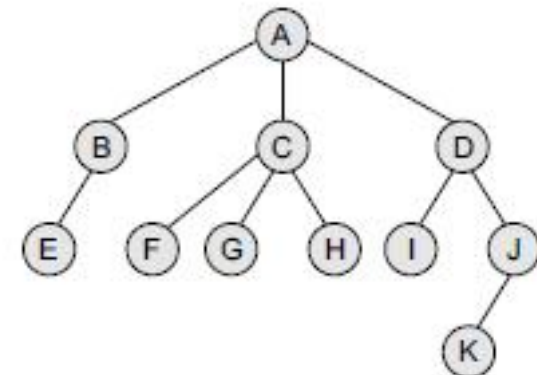
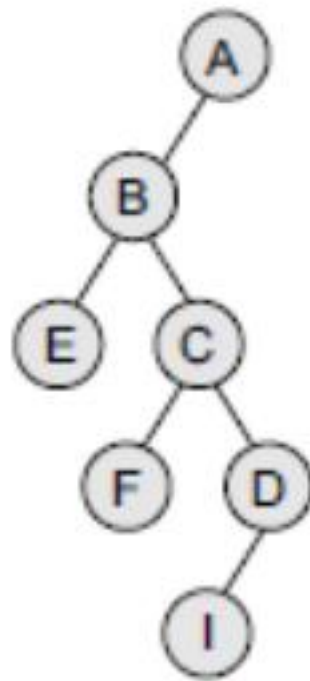
CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 4: Maintenant traitons le nœud C. Le fils gauche de C est F (le fils plus à gauche) et son fils droit est D (frère droit dans l'arbre général).



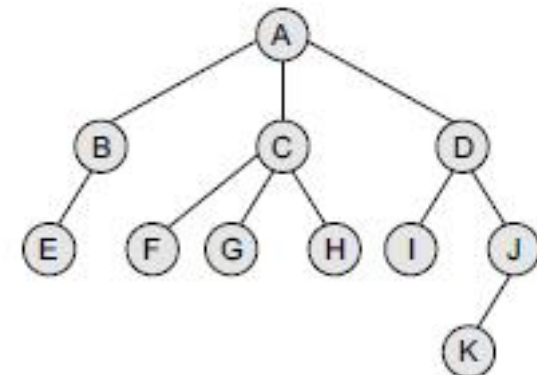
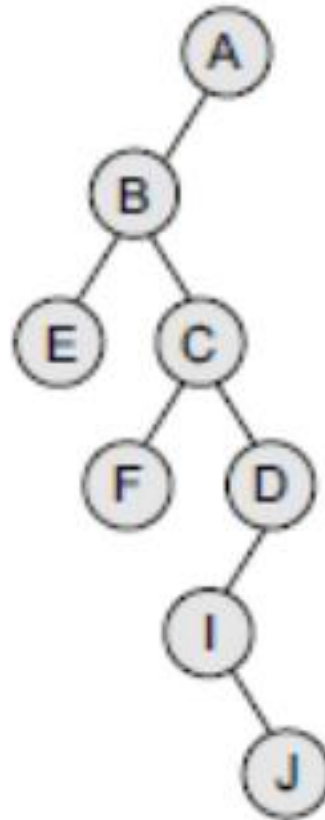
CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 5: Maintenant traitons le nœud D. Fils gauche de D est I (le fils le plus à gauche). Il n'y aura pas de fils droit de D parce qu'il n'a pas de frère droit dans l'arbre général.



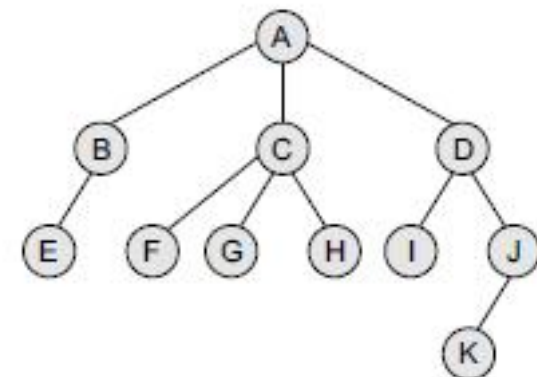
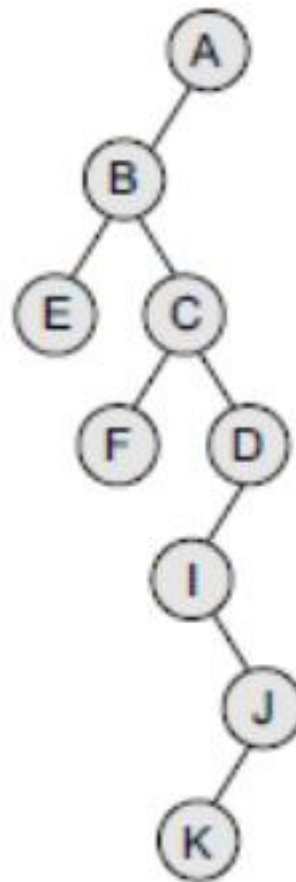
CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 6: Maintenant, traitons le nœud I. Il n'y aura pas de fils de I dans l'arbre binaire parce que I n'a pas de fils dans l'arbre général. Cependant, I a un frère droit J, alors J sera ajouté comme le fils droit de I.



CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 7: Maintenant, traitons le nœud J. Le fils gauche de J est K (fils le plus à gauche). Il n'y aura pas de fils droit de J parce qu'il n'a pas de frère droit dans l'arbre général.



CRÉER UN ARBRE BINAIRE À PARTIR D'UN ARBRE GÉNÉRAL

Étape 8: Maintenant traiter tous les nœuds non transformés (E, F, G, H, K) de la même manière, de sorte que l'arbre binaire qui en résulte peut être donné comme suit.

