

Vocabulaire sur les arbres

- Un **arbre** est une structure de données constituée de **noeuds** reliés entre eux par des **arêtes**.

Vocabulaire sur les arbres

- Un **arbre** est une structure de données constituée de **noeuds** reliés entre eux par des **arêtes**.
- Contrairement aux listes, piles et files qui sont des structures de données **linéaires**, les arbres sont des structures de données **hierarchisées**.

Vocabulaire sur les arbres

- Un **arbre** est une structure de données constituée de **noeuds** reliés entre eux par des **arêtes**.
- Contrairement aux listes, piles et files qui sont des structures de données **linéaires**, les arbres sont des structures de données **hierarchisées**.
- On dit qu'un noeud B est le **fil**s d'un noeud A lorsqu'une arête va du noeud A au noeud B.

Vocabulaire sur les arbres

- Un **arbre** est une structure de données constituée de **noeuds** reliés entre eux par des **arêtes**.
- Contrairement aux listes, piles et files qui sont des structures de données **linéaires**, les arbres sont des structures de données **hierarchisées**.
- On dit qu'un noeud B est le **fil**s d'un noeud A lorsqu'une arête va du noeud A au noeud B.
- Dans un arbre, un seul et unique noeud n'est le fils de personne, c'est la **racine** de l'arbre.

Vocabulaire sur les arbres

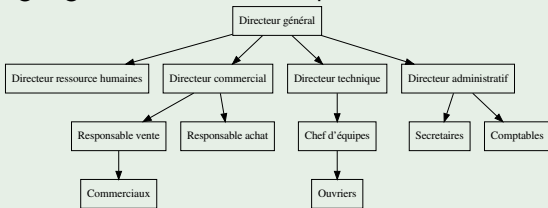
- Un **arbre** est une structure de données constituée de **noeuds** reliés entre eux par des **arêtes**.
- Contrairement aux listes, piles et files qui sont des structures de données **linéaires**, les arbres sont des structures de données **hierarchisées**.
- On dit qu'un noeud B est le **fils** d'un noeud A lorsqu'une arête va du noeud A au noeud B.
- Dans un arbre, un seul et unique noeud n'est le fils de personne, c'est la **racine** de l'arbre.
- Un noeud n'ayant pas de fils s'appelle une **feuille** de l'arbre.

Vocabulaire sur les arbres

- Un **arbre** est une structure de données constituée de **noeuds** reliés entre eux par des **arêtes**.
- Contrairement aux listes, piles et files qui sont des structures de données **linéaires**, les arbres sont des structures de données **hiérarchisées**.
- On dit qu'un noeud B est le **fil**s d'un noeud A lorsqu'une arête va du noeud A au noeud B.
- Dans un arbre, un seul et unique noeud n'est le fils de personne, c'est la **racine** de l'arbre.
- Un noeud n'ayant pas de fils s'appelle une **feuille** de l'arbre.
- On appelle **branche** une suite finie de noeuds partant de la racine vers une feuille.

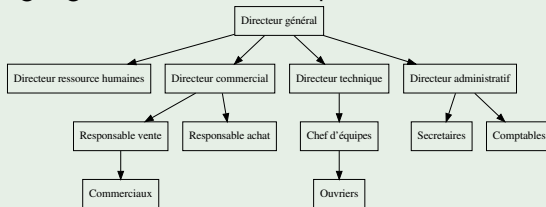
Exemple

On représente l'organigramme d'une société par un arbre :



Exemple

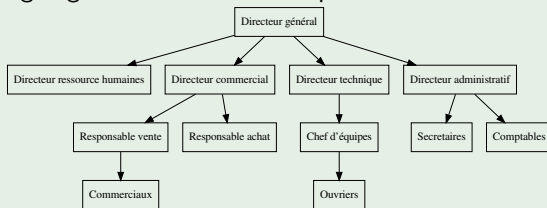
On représente l'organigramme d'une société par un arbre :



- Citer trois noeuds de cet arbre.

Exemple

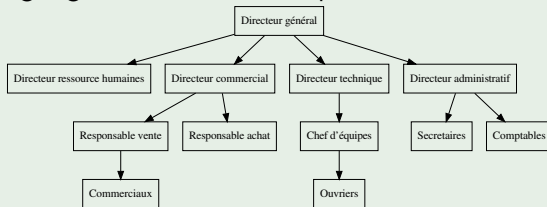
On représente l'organigramme d'une société par un arbre :



- Citer trois noeuds de cet arbre.
- Donner la racine de cet arbre.

Exemple

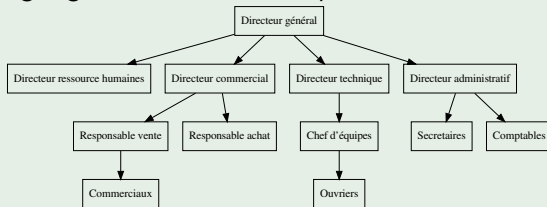
On représente l'organigramme d'une société par un arbre :



- Citer trois noeuds de cet arbre.
- Donner la racine de cet arbre.
- Donner les feuilles de cet arbre.

Exemple

On représente l'organigramme d'une société par un arbre :



- Citer trois noeuds de cet arbre.
- Donner la racine de cet arbre.
- Donner les feuilles de cet arbre.
- Donner une branche de longueur 3 dans cet arbre

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.

Exemple

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.

Exemple

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.

Exemple

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.
Différentes définitions existent pour la hauteur d'un arbre, on considère parfois que la hauteur est le nombre maximal d'arête que peut avoir une branche.

Exemple

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.
Différentes définitions existent pour la hauteur d'un arbre, on considère parfois que la hauteur est le nombre maximal d'arête que peut avoir une branche.
- L'**arité** d'un arbre est le nombre maximal de fils qu'un noeud peut avoir.

Exemple

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.
Différentes définitions existent pour la hauteur d'un arbre, on considère parfois que la hauteur est le nombre maximal d'arête que peut avoir une branche.
- L'**arité** d'un arbre est le nombre maximal de fils qu'un noeud peut avoir.
On parle aussi de l'arité (ou degré) d'un noeud, il s'agit alors du nombre de fils de ce noeud

Exemple

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.
Différentes définitions existent pour la hauteur d'un arbre, on considère parfois que la hauteur est le nombre maximal d'arête que peut avoir une branche.
- L'**arité** d'un arbre est le nombre maximal de fils qu'un noeud peut avoir.
On parle aussi de l'arité (ou degré) d'un noeud, il s'agit alors du nombre de fils de ce noeud

Exemple

- Quelle est la taille de l'arbre représentant l'organigramme d'une société ci-dessus ?

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.
Différentes définitions existent pour la hauteur d'un arbre, on considère parfois que la hauteur est le nombre maximal d'arête que peut avoir une branche.
- L'**arité** d'un arbre est le nombre maximal de fils qu'un noeud peut avoir.
On parle aussi de l'arité (ou degré) d'un noeud, il s'agit alors du nombre de fils de ce noeud

Exemple

- Quelle est la taille de l'arbre représentant l'organigramme d'une société ci-dessus ?
- Sa hauteur ?

Définitions

- La **taille** d'un arbre est le nombre de noeuds de cet arbre.
L'arbre vide n'a aucun noeud, sa taille est 0.
- La **hauteur** d'un arbre est le nombre de noeud maximal qu'une branche peut avoir.
Différentes définitions existent pour la hauteur d'un arbre, on considère parfois que la hauteur est le nombre maximal d'arête que peut avoir une branche.
- L'**arité** d'un arbre est le nombre maximal de fils qu'un noeud peut avoir.
On parle aussi de l'arité (ou degré) d'un noeud, il s'agit alors du nombre de fils de ce noeud

Exemple

- Quelle est la taille de l'arbre représentant l'organigramme d'une société ci-dessus ?
- Sa hauteur ?
- Son arité ?

Arbre binaire

Arbre binaire

- On appelle **arbre binaire** un arbre dans lequel tous les noeuds ont au maximum deux fils.

Arbre binaire

- On appelle **arbre binaire** un arbre dans lequel tous les noeuds ont au maximum deux fils.
De façon équivalent, on peut dire qu'un arbre binaire est un arbre d'arité 2.

Arbre binaire

- On appelle **arbre binaire** un arbre dans lequel tous les noeuds ont au maximum deux fils.
De façon équivalente, on peut dire qu'un arbre binaire est un arbre d'arité 2.
- Chaque noeud ayant au plus deux fils, dans un arbre binaire on peut considérer le **sous arbre droit** et le **sous arbre gauche** d'un noeud. Chacun de ces sous arbres étant lui-même un arbre binaire pouvant être vide (noté Δ).

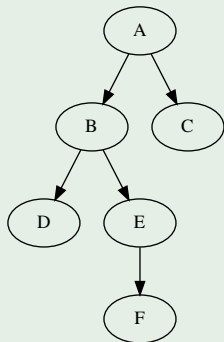
Arbre binaire

- On appelle **arbre binaire** un arbre dans lequel tous les noeuds ont au maximum deux fils.
De façon équivalente, on peut dire qu'un arbre binaire est un arbre d'arité 2.
- Chaque noeud ayant au plus deux fils, dans un arbre binaire on peut considérer le **sous arbre droit** et le **sous arbre gauche** d'un noeud. Chacun de ces sous arbres étant lui-même un arbre binaire pouvant être vide (noté Δ).
On obtient donc une définition récursive d'un arbre binaire :

Arbre binaire

- On appelle **arbre binaire** un arbre dans lequel tous les noeuds ont au maximum deux fils.
De façon équivalent, on peut dire qu'un arbre binaire est un arbre d'arité 2.
- Chaque noeud ayant au plus deux fils, dans un arbre binaire on peut considérer le **sous arbre droit** et le **sous arbre gauche** d'un noeud. Chacun de ces sous arbres étant lui-même un arbre binaire pouvant être vide (noté Δ). On obtient donc une définition récursive d'un arbre binaire :
Un arbre binaire est soit un arbre vide (Δ) soit un triplet $(noeud, sag, sad)$ où *sag* et *sad* sont des arbres binaires.

Exemple



- Vérifier rapidement que cet arbre est binaire.
- Compléter l'écriture de cet arbre binaire sous forme de triplet :

$(A, \text{sag}(A), \text{sad}(A))$

$(A, (B, \text{sag}(B), \text{sad}(B)), (C, \Delta, \Delta))$

.....

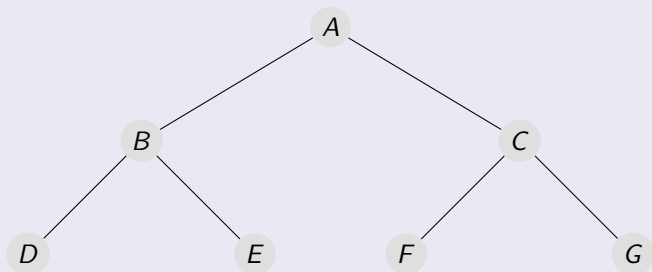
- Dessiner l'arbre binaire qui s'écrit sous forme de triplet :
 $(A, (B, (C, (D, \Delta, \Delta), \Delta), (E, \Delta, \Delta)))$.

Quelques cas particuliers

- Un arbre binaire est dit *complet* lorsque tous les niveaux sont remplis :

Quelques cas particuliers

- Un arbre binaire est dit *complet* lorsque tous les niveaux sont remplis :
Exemple d'arbre binaire complet de hauteur 3.



Relation entre hauteur et taille

En notant n la taille et h la hauteur d'un arbre binaire, pour $n \geq 2$, on a la relation suivante :

Relation entre hauteur et taille

En notant n la taille et h la hauteur d'un arbre binaire, pour $n \geq 2$, on a la relation suivante :

$$h \leq n \leq 2^h - 1$$

Parcours d'un arbre

On peut parcourir un arbre binaire :

Parcours d'un arbre

On peut parcourir un arbre binaire :

- En largeur, cela revient à lister les noeuds par ordre croissant de profondeur et de gauche à droite

Parcours d'un arbre

On peut parcourir un arbre binaire :

- En largeur, cela revient à lister les noeuds par ordre croissant de profondeur et de gauche à droite
L'implémentation de ce parcours peut se faire à l'aide d'une file dans laquelle on stocke les noeuds restants à parcourir. A chaque fois qu'on traite un noeud, on le defile et on enfile ses fils (voir la fiche d'activité).
- En profondeur, on tire alors partie de la structure récursive des arbres. Pour parcourir l'arbre $T = (e, sag, sad)$ on doit relancer le parcours sur *sag* et *sad*. On distingue alors trois parcours suivant que *e* est affiché avant, entre ou après *sag* et *sad* :

Parcours d'un arbre

On peut parcourir un arbre binaire :

- En largeur, cela revient à lister les noeuds par ordre croissant de profondeur et de gauche à droite

L'implémentation de ce parcours peut se faire à l'aide d'une file dans laquelle on stocke les noeuds restants à parcourir. A chaque fois qu'on traite un noeud, on le defile et on enfile ses fils (voir la fiche d'activité).

- En profondeur, on tire alors partie de la structure récursive des arbres. Pour parcourir l'arbre $T = (e, sag, sad)$ on doit relancer le parcours sur *sag* et *sad*. On distingue alors trois parcours suivant que *e* est affiché avant, entre ou après *sag* et *sad* :
 - Dans le parcours préfixé, *e* est affiché avant de parcourir *sag* et *sad*.

Parcours d'un arbre

On peut parcourir un arbre binaire :

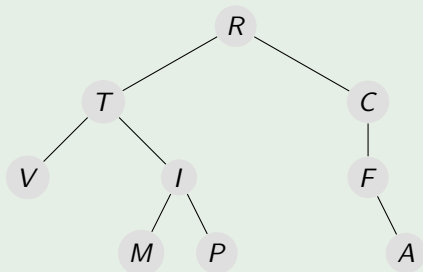
- En largeur, cela revient à lister les noeuds par ordre croissant de profondeur et de gauche à droite
L'implémentation de ce parcours peut se faire à l'aide d'une file dans laquelle on stocke les noeuds restants à parcourir. A chaque fois qu'on traite un noeud, on le defile et on enfile ses fils (voir la fiche d'activité).
- En profondeur, on tire alors partie de la structure récursive des arbres. Pour parcourir l'arbre $T = (e, sag, sad)$ on doit relancer le parcours sur *sag* et *sad*. On distingue alors trois parcours suivant que *e* est affiché avant, entre ou après *sag* et *sad* :
 - Dans le parcours préfixé, *e* est affiché avant de parcourir *sag* et *sad*.
 - Dans le parcours infixé, *e* est affiché après le parcours de *sag* mais avant celui de *sad*.

Parcours d'un arbre

On peut parcourir un arbre binaire :

- En largeur, cela revient à lister les noeuds par ordre croissant de profondeur et de gauche à droite
L'implémentation de ce parcours peut se faire à l'aide d'une file dans laquelle on stocke les noeuds restants à parcourir. A chaque fois qu'on traite un noeud, on le defile et on enfile ses fils (voir la fiche d'activité).
- En profondeur, on tire alors partie de la structure récursive des arbres. Pour parcourir l'arbre $T = (e, sag, sad)$ on doit relancer le parcours sur *sag* et *sad*. On distingue alors trois parcours suivant que *e* est affiché avant, entre ou après *sag* et *sad* :
 - Dans le parcours préfixé, *e* est affiché avant de parcourir *sag* et *sad*.
 - Dans le parcours infixé, *e* est affiché après le parcours de *sag* mais avant celui de *sad*.
 - Dans le parcours suffixé, *e* est affiché après le parcours de *sag* et *sad*

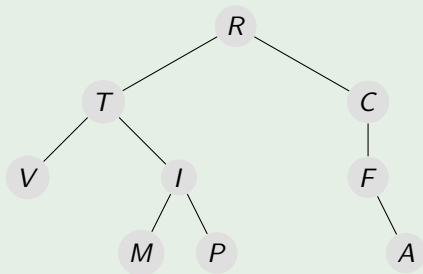
Exemple



Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur

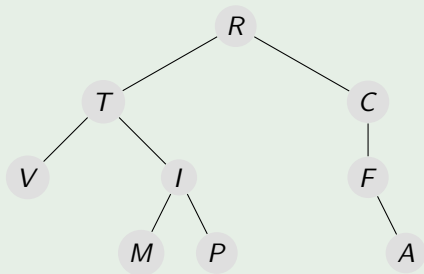
Exemple



Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur
- En profondeur préfixé

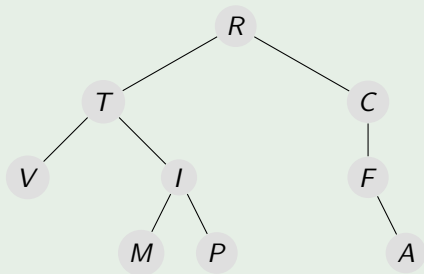
Exemple



Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur
- En profondeur préfixé
- En profondeur infixé

Exemple



Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur
- En profondeur préfixé
- En profondeur infixé
- En profondeur suffixé