

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 **hiérarchisées**.

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** 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 **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.
- On appelle **branche** une suite finie de noeuds partant de la racine vers une feuille.

Définitions

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

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.

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.

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.

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.

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

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 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 :

Arbre binaire

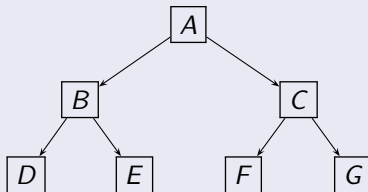
- On appelle **arbre binaire** un arbre dans lequel tous les noeuds ont au maximum deux fils.
De façon équivalent, on peut dire qu'un arbre binaire est un arbre d'arité 2.
- Chaque noeud ayant au plus deux fils, dans un arbre binaire on peut considérer le **sous arbre droit** et le **sous arbre gauche** 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.

Cas particuliers

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

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 :

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

C22 Algorithmes sur les arbres

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 défile et on enfile ses fils.

- En profondeur, on tire alors partie de la structure récursive des arbres. Pour parcourir l'arbre $T = (racine, sag, sad)$ on doit relancer le parcours sur *sag* et *sad*. On distingue alors trois parcours suivant que *racine* 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 défile et on enfile ses fils.

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

C22 Algorithmes sur les arbres

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

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

C22 Arbres binaires de Recherche

Arbre binaire de recherche

Un arbre binaire **de recherche** (noté ABR), est un arbre binaire tel que :

Arbre binaire de recherche

Un arbre binaire **de recherche** (noté ABR), est un arbre binaire tel que :

- Les étiquettes des noeuds, appelées **clé** sont toutes comparables entre elles.

Arbre binaire de recherche

Un arbre binaire **de recherche** (noté ABR), est un arbre binaire tel que :

- Les étiquettes des noeuds, appelées **clé** sont toutes comparables entre elles.

Arbre binaire de recherche

Un arbre binaire **de recherche** (noté ABR), est un arbre binaire tel que :

- Les étiquettes des noeuds, appelées **clé** sont toutes comparables entre elles.
Par exemple, les étiquettes sont toutes des nombres ou encore des chaînes de caractères (comparées par ordre alphabétique).

Arbre binaire de recherche

Un arbre binaire **de recherche** (noté ABR), est un arbre binaire tel que :

- Les étiquettes des noeuds, appelées **clé** sont toutes comparables entre elles.
Par exemple, les étiquettes sont toutes des nombres ou encore des chaînes de caractères (comparées par ordre alphabétique).
- Pour tous les noeuds l'ensemble des clés présentes dans le sous arbre gauche (resp. droit) sont strictement inférieures (resp. supérieures) à la clé du noeud.

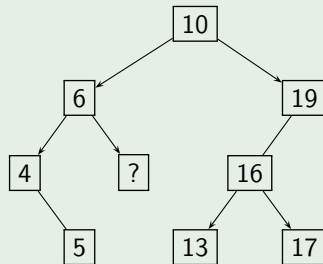
Arbre binaire de recherche

Un arbre binaire **de recherche** (noté ABR), est un arbre binaire tel que :

- Les étiquettes des noeuds, appelées **clé** sont toutes comparables entre elles.
Par exemple, les étiquettes sont toutes des nombres ou encore des chaînes de caractères (comparées par ordre alphabétique).
- Pour tous les noeuds l'ensemble des clés présentes dans le sous arbre gauche (resp. droit) sont strictement inférieures (resp. supérieures) à la clé du noeud.
Par souci de simplicité, on admettra que les clés sont uniques dans un ABR ce qui permet d'éviter le cas de clés égales

C22 Arbres binaires de Recherche

Exemple



Cet arbre est-il un ABR si la clé manquante est 2 ? 9 ? 12 ?

Recherche dans un ABR

- La recherche d'un élément dans un ABR a pour complexité la hauteur de cet arbre. En effet, on descend d'un niveau dans l'arbre à chaque étape de la recherche en choisissant d'aller à gauche ou à droite suivante que l'élément recherché est plus petit ou plus grand que le noeud parcouru.

Recherche dans un ABR

- La recherche d'un élément dans un ABR a pour complexité la hauteur de cet arbre. En effet, on descend d'un niveau dans l'arbre à chaque étape de la recherche en choisissant d'aller à gauche ou à droite suivante que l'élément recherché est plus petit ou plus grand que le noeud parcouru.
- Par conséquent, si l'arbre est dégénéré, la hauteur est égale au nombre de noeuds et l'algorithme équivaut à la recherche dans une liste.

Recherche dans un ABR

- La recherche d'un élément dans un ABR a pour complexité la hauteur de cet arbre. En effet, on descend d'un niveau dans l'arbre à chaque étape de la recherche en choisissant d'aller à gauche ou à droite suivante que l'élément recherché est plus petit ou plus grand que le noeud parcouru.
- Par conséquent, si l'arbre est dégénéré, la hauteur est égale au nombre de noeuds et l'algorithme équivaut à la recherche dans une liste.
- Si l'arbre est complet par contre la complexité est logarithmique et équivaut à une recherche dichotomique dans une liste triée.