

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*.

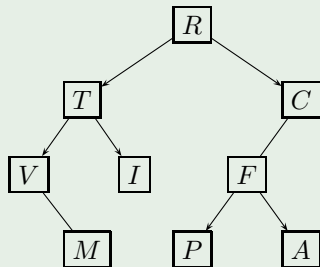
Parcours d'un arbre

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*

C10 Algorithmes sur les arbres

Exemple

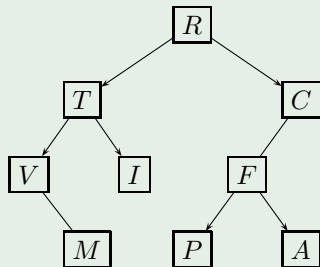


Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur

C10 Algorithmes sur les arbres

Exemple

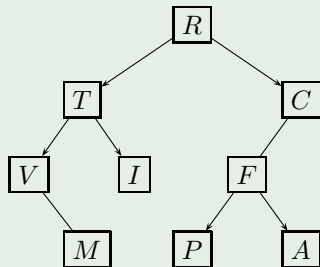


Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur
- En profondeur préfixé

C10 Algorithmes sur les arbres

Exemple

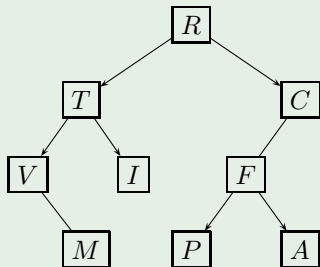


Donner l'ordre des noeuds lorsqu'on parcourt l'arbre ci-dessus :

- En largeur
- En profondeur préfixé
- En profondeur infixé

C10 Algorithmes sur les arbres

Exemple



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

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

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.

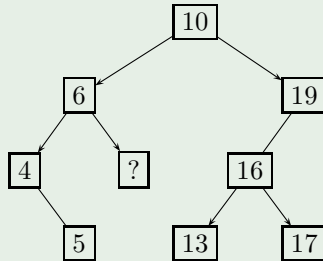
Arbre binaire de recherche

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

C10 Algorithmes sur les arbres

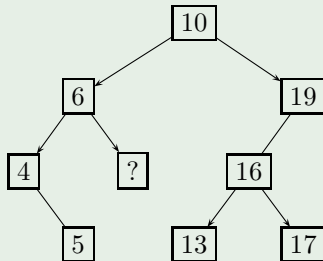
Exemple



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

C10 Algorithmes sur les arbres

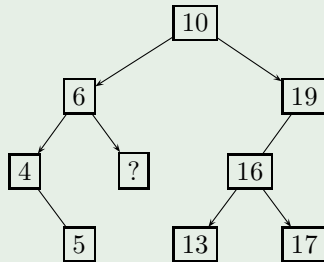
Exemple



- Cet arbre est-il un ABR si la clé manquante est 2 ? 9 ? 12 ?
- On suppose que la clé manquante est 9. Proposer une nouvelle valeur pour le noeud de clé 16 de façon à ce que cet arbre reste un ABR.

C10 Algorithmes sur les arbres

Exemple



- Cet arbre est-il un ABR si la clé manquante est 2 ? 9 ? 12 ?
- On suppose que la clé manquante est 9. Proposer une nouvelle valeur pour le noeud de clé 16 de façon à ce que cet arbre reste un ABR.
- Proposer une valeur pour le noeud de clé 16 de façon à ce que cet arbre ne soit pas un ABR.

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.