

Arbres Binaires

Dans tout ce qui suit A est un arbre binaire.

Implantation par chaînage fils gauche - fils droit

Un nœud est une cellule contenant un champ FG , un champ $Etiquette$ et un champ FD . Un arbre est un pointeur sur une cellule de ce type.

Si n est un nœud, $n \rightarrow FG$ pointe sur le fils gauche de n , $n \rightarrow FD$ pointe sur le fils droit, et $n \rightarrow Etiquette$ contient l'étiquette (ou la valeur) de n .

Algorithme 1 Fonction $ParcoursPrefixe(n)$

```
si  $n \neq \text{NULL}$  alors
  Afficher( $n \rightarrow Etiquette$ );
  ParcoursPrefixe( $n \rightarrow FG$ );
  ParcoursPrefixe( $n \rightarrow FD$ )
fin si
```

Appel avec $ParcoursPrefixe(A)$.

Algorithme 2 Fonction $ParcoursInterne(n)$

```
si  $n \neq \text{NULL}$  alors
  ParcoursInterne( $n \rightarrow FG$ );
  Afficher( $n \rightarrow Etiquette$ );
  ParcoursInterne( $n \rightarrow FD$ )
fin si
```

Appel avec $ParcoursInterne(A)$.

Algorithme 3 Fonction $ParcoursSuffixe(n)$

```
si  $n \neq \text{NULL}$  alors
  ParcoursSuffixe( $n \rightarrow FG$ );
  ParcoursSuffixe( $n \rightarrow FD$ );
  Afficher( $n \rightarrow Etiquette$ )
fin si
```

Appel avec $ParcoursSuffixe(A)$.

Parcours par niveau

On utilise une queue de pointeurs Q , et les fonctions définies dans le cours sur les queues.

Algorithme 4 Fonction ParcoursNiveau(A)

```
si  $A \neq \text{NULL}$  alors
  Initialiser( $Q$ );
  AjouterQueue( $A, Q$ )
  tant que non Vide( $Q$ ) faire
     $p := \text{Tete}(Q)$ ;
    EnlQueue( $Q$ );
    Afficher( $p \rightarrow \text{Etiquette}$ );
    si  $p \rightarrow FG \neq \text{NULL}$  alors
      AjouterQueue( $p \rightarrow FG, Q$ );
    fin si
    si  $p \rightarrow FD \neq \text{NULL}$  alors
      AjouterQueue( $p \rightarrow FD, Q$ );
    fin si
  fin tant que
fin si
```

Hauteur d'un arbre binaire

Algorithme 5 Fonction Hauteur(A)

```
si  $A = \text{NULL}$  alors
  RETURN (-1)
sinon
  RETURN (1 + max(Hauteur( $A \rightarrow FG$ ), Hauteur( $A \rightarrow FD$ )))
fin si
```
