

Pour tous les types présentés ici, on suppose le type `t_element` (le type des contenus des nœuds) défini.

1 Implémentation dynamique

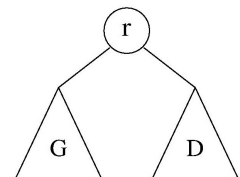
Cette représentation est quasiment un calque de la structure d'un arbre binaire. Chaque nœud est représenté par un enregistrement qui contient un élément (l'étiquette) et deux pointeurs : un vers le fils gauche et l'autre sur le fils droit.

Calque de la structure récursive

Un arbre sera représenté par un unique pointeur :

arbre vide : le pointeur NUL

arbre non vide : un pointeur sur l'enregistrement représentant la racine.



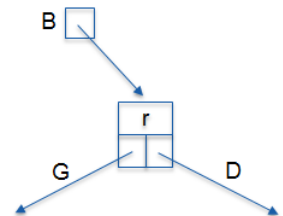
Le type

types

```
t_arbreBinaire = ↑ t_noeud
t_noeud = enregistrement
    t_element    elt
    t_arbreBinaire fg, fd
fin enregistrement t_noeud
```

variables

```
t_arbreBinaire B
```



Implémentation des opérations

$B : \text{ArbreBinaire}$	$B : \text{t_arbreBinaire}$
arbre-vide	NUL
$\text{racine}(B)$	$B \uparrow$
$\text{contenu}(\text{racine}(B))$	$B \uparrow . \text{elt}$
$g(B)$	$B \uparrow . \text{fg}$
$d(B)$	$B \uparrow . \text{fd}$
$B \leftarrow \langle _, _, _ \rangle$	$\text{allouer}(B)$

2 Implémentations statiques

2.1 "Faux-pointeurs"

Comme dans l'implémentation dynamique, chaque nœud de l'arbre est représenté par un enregistrement. Ceux-ci sont dans un tableau (1 dimension, un vecteur), et les pointeurs sont remplacés par des entiers indiquant la position des fils dans le tableau (le pointeur NUL sera remplacé par 0 ou -1).

Un arbre sera représenté par un enregistrement contenant :

- le tableau contenant les éléments ;
- un entier, représentant la position de la racine dans le tableau.

Algorithmique

Arbres binaires : implémentations

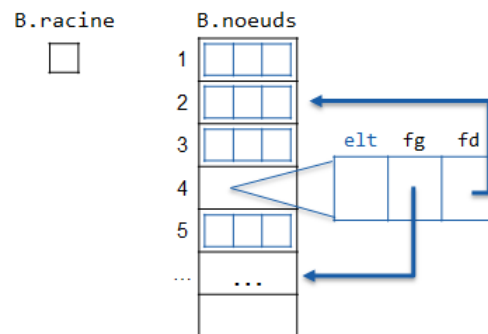
Le type

constantes

Nbmax = ...

types

```
t_noeud = enregistrement
  t_element  elt
  entier     fg, fd
fin enregistrement t_noeud
t_vectnoeuds = Nbmax t_noeud
t_arbre = enregistrement
  t_vectnoeuds noeuds
  entier       racine
fin enregistrement t_arbre
```



variables

t_arbreBinaire B

Implémentation des opérations

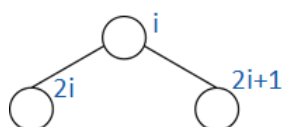
$B : \text{ArbreBinaire}$	$B : \text{t_arbreBinaire}$
$B = \text{arbre-vide}$	$B.\text{racine} = 0$
$\text{racine}(B)$	$B.\text{noeuds}[B.\text{racine}]$
$\text{contenu}(\text{racine}(B))$	$B.\text{noeuds}[B.\text{racine}].\text{elt}$
$g(B)$	$B.\text{noeuds}[B.\text{racine}].\text{fg}$
$d(B)$	$B.\text{noeuds}[B.\text{racine}].\text{fd}$

2.2 Un simple tableau

Numérotation hiérarchique

On peut utiliser un simple vecteur (tableau à une dimension) pour représenter un arbre binaire étiqueté. Il suffit de stocker chaque valeur à la position correspondant au numéro en ordre hiérarchique du nœud la contenant.

Utilisation



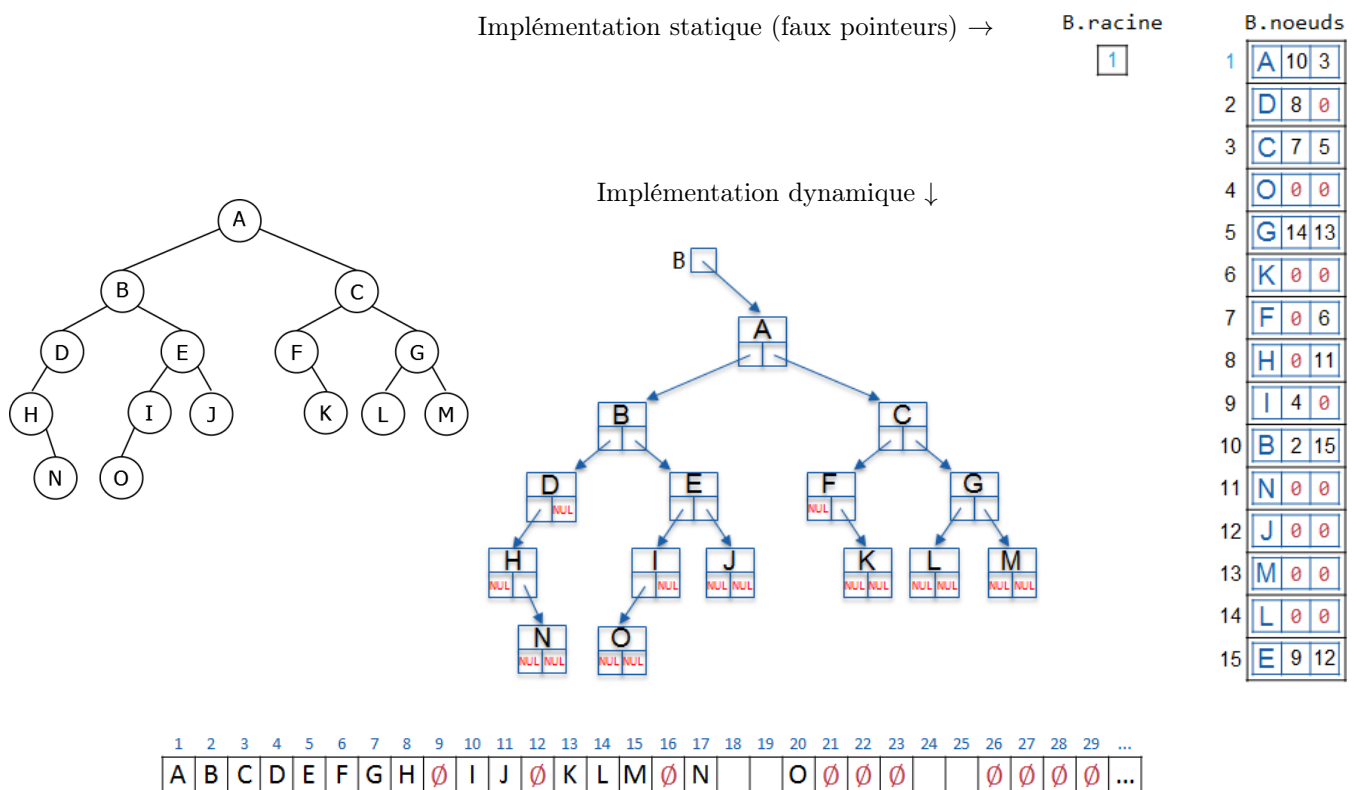
- La racine est en position 1 dans le vecteur.
- Si i est la position du nœud actuel alors :
 - ▷ son fils gauche se trouve à la position $2i$
 - ▷ son fils droit se trouve à la position $2i + 1$
 - ▷ son père (sauf pour la racine de l'arbre), se trouve à la position $i \text{ div } 2$.

Il faut de plus avoir une valeur particulière (\emptyset) pour remplacer la racine de l'arbre vide.

3 Conclusion : comparaison des 3 implémentations

	Dynamique	Statique (faux pointeurs)	Statique avec numérotation hiérarchique
Mémoire	La plus efficace : un enregistrement par nœud	Moyen si tableau statique : la place occupée est la même quelque soit la taille de l'arbre	La plus efficace sur arbre parfait, la pire sur arbre quelconque (prévoir jusqu'à 2^n cases pour un arbre dégénéré de n nœuds)
Parcours	Simple : il suffit de "suivre" les pointeurs	Simple : on suit également les liens (en emportant l'entier correspondant à la position du nœud actuel dans le tableau)	Simple : utiliser les règles de la numérotation hiérarchique Seule implémentation permettant l'accès au père immédiatement
Modifications	Simple comme toute implémentation dynamique	La suppression peut ne pas poser de problème, par contre l'insertion peut nécessiter un parcours pour trouver une "place libre"	Pas du tout adapté sauf dans des cas très particuliers (arbres parfaits, modification sur dernier niveau)

Annexe : Schémas



Implémentation statique avec numérotation hiérarchique