

I. Arbres et récursivité

Enregistrement *nœud*

valeur : entier,
fils : tableau de liens de nœuds

Fin enregistrement

Procédure hauteur(racine)

Spécification :

Entrée : un lien sur un nœud,
Sortie : un entier

Variables :

h_{\max} : entier

$h_{\max} \leftarrow 0$

Si racine $\neq \emptyset$ alors

Pour $i \in \llbracket 0, \text{len}(\text{nœud} \rightarrow \text{fils}) - 1 \rrbracket$ faire

$h_{\max} \leftarrow \max(h_{\max}, \text{hauteur}(\text{nœud} \rightarrow \text{fils}[i]))$

Fin pour

$h_{\max} \leftarrow 1 + h_{\max}$

Fin si

Renvoyer h_{\max}

Procédure nb_nœuds(racine)

Spécification :

Entrée : un lien sur un nœud,
Sortie : un entier

Variables :

total : entier

total \leftarrow 0

Si racine $\neq \emptyset$ alors

Pour $i \in \llbracket 0, \text{len}(\text{nœud} \rightarrow \text{fils}) - 1 \rrbracket$ **faire**

 total \leftarrow total + nb_nœuds(nœud \rightarrow fils[i])

Fin pour

total \leftarrow 1 + total

Fin si

Renvoyer total

Procédure nb_feuilles(racine)

Spécification :

Entrée : un lien sur un nœud,
Sortie : un entier

Variables :

total : entier

total \leftarrow 0

Si racine $\neq \emptyset$ alors

Pour $i \in \llbracket 0, \text{len}(\text{nœud} \rightarrow \text{fils}) - 1 \rrbracket$ **faire**

 total \leftarrow total + nb_feuilles(nœud \rightarrow fils[i])

Fin pour

total \leftarrow max(1, total)

Fin si

Renvoyer total

Procédure affiche_arbre(racine)

Spécification :

Entrée : un lien sur un nœud,

Sortie : rien

Si racine $\neq \emptyset$ **alors**

Afficher nœud \rightarrow valeur

Pour $i \in \llbracket 0, \text{len}(\text{nœud} \rightarrow \text{fils}) - 1 \rrbracket$ **faire**

 affiche_arbre(nœud \rightarrow fils[i])

Fin pour

Fin si

I.1. Réservation de salle

Enregistrement nœud

début : entier,

fin : entier,

f_g : lien sur nœud,

f_d : lien sur nœud

Fin enregistrement

Procédure peut_ajouter(début, fin, racine)

Spécification :

Entrée : deux entiers, et un lien sur un nœud,

Sortie : un booléen

Si racine = \emptyset **alors**

Renvoyer VRAI

Sinon si racine \rightarrow fin < début **alors**

Renvoyer peut_ajouter(début, fin, racine $\rightarrow f_d$)

Sinon si racine \rightarrow début > fin **alors**

Renvoyer peut_ajouter(début, fin, racine $\rightarrow f_g$)

Sinon

Renvoyer FAUX

Fin si

Procédure réserve(début, fin, racine)

Spécification :

Entrée : deux entiers, et un lien sur un nœud,

Sortie : un lien sur un nœud

Si racine = \emptyset alors

$n \leftarrow \text{nouveau}()$

$n \rightarrow f_g \leftarrow \emptyset$

$n \rightarrow f_d \leftarrow \emptyset$

$n \rightarrow \text{début} \leftarrow \text{début}$

$n \rightarrow \text{fin} \leftarrow \text{fin}$

Renvoyer n

Sinon si racine \rightarrow début \leq début alors

$\text{racine} \rightarrow f_g \leftarrow \text{réserve}(\text{début}, \text{fin}, \text{racine} \rightarrow f_g)$

Renvoyer racine

Sinon

$\text{racine} \rightarrow f_d \leftarrow \text{réserve}(\text{début}, \text{fin}, \text{racine} \rightarrow f_d)$

Renvoyer racine

Fin si

Procédure annule(début, fin, racine)

Spécification :

Entrée : deux entiers, et un lien sur un nœud,

Sortie : un lien sur un nœud

Si racine \rightarrow début = début alors

$\text{racine} \leftarrow \text{supprime_racine}(\text{racine})$

Sinon si racine \rightarrow début < début alors

$\text{racine} \rightarrow f_g \leftarrow \text{annule}(\text{début}, \text{fin}, \text{racine} \rightarrow f_g)$

Sinon

$\text{racine} \rightarrow f_d \leftarrow \text{annule}(\text{début}, \text{fin}, \text{racine} \rightarrow f_d)$

Fin si

Retourne racine

Procédure trouve_maximum(racine)

Spécification :

| *Entrée* : un lien sur un nœud,

| *Sortie* : deux lien sur un nœud

Variables :

| temp : un lien sur un nœud

Si racine = \emptyset **alors**

| **Renvoyer** racine

Sinon

| $(temp, racine \rightarrow f_g) \leftarrow$ trouve_maximum($racine \rightarrow f_g$)

| **Si** temp = \emptyset **alors**

| | temp \leftarrow racine

| **Fin si**

| **Renvoyer** temp

Fin si

Procédure `supprime_racine(racine)`

Spécification :

Entrée : un lien sur un nœud,
Sortie : un lien sur un nœud

Variables :

temp : lien sur un nœud

Si $\text{racine} \rightarrow f_g = \emptyset$ alors

temp \leftarrow $\text{racine} \rightarrow f_d$
libère(racine)

Renvoyer temp

Sinon si $\text{racine} \rightarrow f_d = \emptyset$ alors

temp \leftarrow $\text{racine} \rightarrow f_g$
libère(racine)

Renvoyer temp

Sinon

on le remplace par l'élément maximal de f_g

$(\text{temp}, \text{racine} \rightarrow f_g) \leftarrow \text{trouve_maximum}(\text{racine} \rightarrow f_g)$

$\text{temp} \rightarrow f_g \leftarrow \text{racine} \rightarrow f_g$

$\text{temp} \rightarrow f_d \leftarrow \text{racine} \rightarrow f_d$

libère(racine)

Renvoyer temp

Fin si

Procédure `afficher(racine)`

Spécification :

Entrée : un lien sur un nœud,
Sortie : rien

Si $\text{racine} \neq \emptyset$ alors

afficher($\text{racine} \rightarrow f_g$)

Afficher « [$\text{racine} \rightarrow \text{début}$]-[$\text{racine} \rightarrow \text{fin}$] »

afficher($\text{racine} \rightarrow f_d$)

Fin si

II. Enracinement

L'opération de rotation à gauche (resp. à droite) conserve les propriétés de l'ABR.