***Comment supprimer une valeur contenue dans un nœud qui a deux fils***

échange\_suppr (a : arbre)

/\* hypothèse : le nœud pointé par a possède deux fils\*/

donnée : a arbre, ce pointeur pointe sur le nœud qui contient une valeur à supprimer

variable locale cour : arbre

variable locale père : arbre

début

/\*recherche du nœud qui contient la plus petite valeur du sous-arbre droit de a\*/

cour 🡨 a 🡪 fd

père 🡨a

tant que cour🡪fg <> nul faire

père🡨cour

cour 🡨 cour 🡪fg

fin

/\* recopier la plus petite valeur à la place de la valeur à supprimer\*/

a🡪info 🡨 cour 🡪info

/\* attacher le fils droit du cour au parent du cour en tant que son fils gauche\*/

père🡪fg 🡨 cour 🡪 fd

libérer (cour)

fin

***Supprimer une valeur dans un arbre***

Suppression\_itérative (a : arbre, e : T) : booléen

Donnée modifiée a : arbre

Donnée e : T

Variable locale cour, père : arbre

résultat de type booléen

début

Si a <>nul alors

/\* suppression de la racine \*/

si e = a🡪 info alors

père 🡨a

/\* l’absence du fils gauche ou de deux fils\*/

si a🡪fg = nul alors

a 🡨 a🡪fd

libérer père

retourner vrai

fin

/\* l’absence du fils droit\*/

si a🡪fd = nul alors

a 🡨 a🡪fg

libérer père

retourner vrai

fin

/\* les deux fils sont présents \* :

échange\_suppr (a)

retourner vrai

fin

/\*suppression d’une valeur qui n’est pas contenue à la racine\*/

père 🡨a

si a🡪info > e alors

cour 🡨 a 🡪fg

sinon

cour 🡨 a🡪fd

fin

tant que cour <> nul faire

/\* on a trouvé le nœud qui contient la valeur à supprimer\*/

si cour 🡪 info = e alors

/\* l’absence du fils gauche ou de deux fils\*/

si cour 🡪 fg = nul alors

/\* vérifier de quelle côte il faut attacher le fils droit du nœud à supprimer au père du nœud à supprimer\*/

si père 🡪info > e alors

père 🡪 fg 🡨cour 🡪fd

sinon

père 🡪 fd 🡨cour 🡪fd

fin

libérer (cour)

retourner vrai

fin

/\* l’absence du fils droit \*/

si cour 🡪 fd = nul alors

/\* vérifier de quelle côte il faut attacher le fils gauche du nœud à supprimer au père du nœud à supprimer\*/

si père 🡪info > e alors

père 🡪 fg 🡨cour 🡪fg

sinon

père 🡪 fd 🡨cour 🡪fg

fin

libérer (cour)

retourner vrai

fin

/\* les deux fils sont présents \*/

échange\_suppr (cour)

retourner vrai

fin

/\* on continue de chercher \*/

père 🡨 cour

si cour 🡪 info > e alors

cour 🡨cour 🡪 fg

sinon

cour 🡨 cour🡪 fd

fin

fin

/\* la valeur n’a pas été trouvée \*/

retourner faux

fin

/\* l’arbre était vide au départ \*/

retourner faux

fin