

### contient()

Pour chercher un élément, on part de la racine et on compare l'élément avec l'élément de la racine.

Si c'est égal, on a trouvé.

Si l'élément est plus petit, on va à gauche.

Si l'élément est plus grand, on va à droite.

S'il n'y a plus moyen d'aller dans la direction où on veut, la recherche échoue.

### insere()

Pour insérer un élément, on parcourt l'arbre comme pour une recherche.

La recherche s'achève quand on ne sait plus aller dans la direction voulue : c'est l'endroit où il faut insérer.

Comme la place est libre, on insère un nouveau nœud avec l'élément à cet endroit.

### supprimeMin()

On cherche le minimum.

On supprime le minimum trouvé !

Le minimum n'a pas de fils gauche.

Le nœud à supprimer est remplacé par la racine de son fils droit (son nœud droit).

### supprime()

On commence par rechercher le nœud contenant l'élément à supprimer.

Si ce nœud n'a pas de fils droit, on le remplace par la racine de son fils gauche (son nœud gauche).

Si ce nœud n'a pas de fils gauche, on le remplace par la racine de son fils droit (son nœud droit).

Si ce nœud a des fils gauche et droit, on supprime le nœud min de son fils droit et on recopie la valeur de ce nœud (min) dans le nœud contenant l'élément à supprimer.