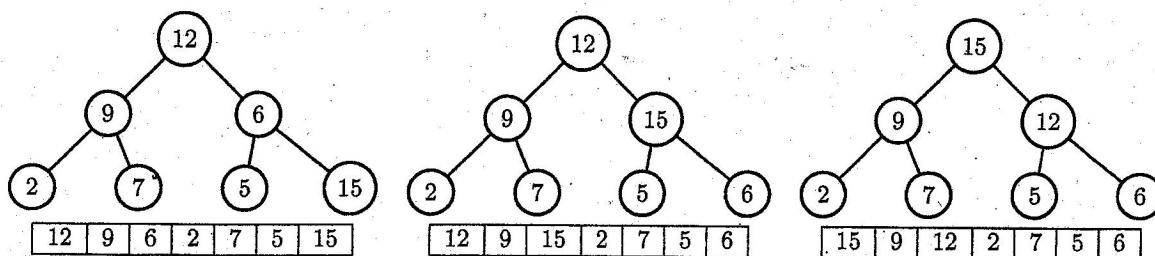


Insertion dans un tas binaire

L'insertion d'un élément dans un tas binaire se ramène à 2 type d'opérations :

1. l'insertion de l'élément dans première cellule vide du tableau codant l'arbre.
2. la *percolation* de cet élément depuis une feuille de l'arbre jusqu'à la racine (si nécessaire). Ces opérations consistent à échanger autant que nécessaire l'élément qui *percole* avec son père *courant* si ce dernier est plus petit que lui.

Dans l'exemple suivant, on ajoute l'élément 15 au tas en 3 étapes :

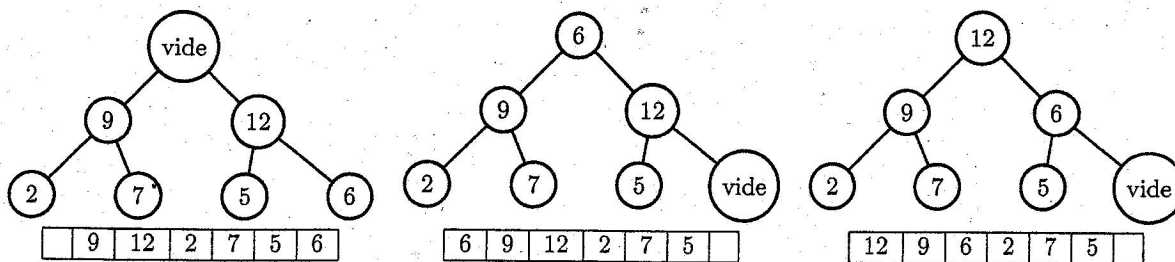


Retirer un élément d'un tas binaire

Pour retirer un élément d'un tas binaire on utilise 3 type d'opérations :

1. on retire la racine de l'arbre (i.e. le premier élément du tableau codant cet arbre) laissant ainsi la première cellule vide et deux sous-arbres.
2. on place la dernière feuille (i.e. le dernier élément du tableau codant l'arbre) à la racine (i.e. le premier élément du tableau codant l'arbre)
3. le *tamissage* de cet élément depuis la racine jusqu'à une feuille (si nécessaire) ; c'est en quelque sorte l'opération inverse de la percolation. Ces opérations consistent à échanger autant que nécessaire l'élément que l'on a placé dans la racine avec le plus grand fils *courant* qui lui est inférieur.

Dans l'exemple suivant, on retire l'élément 15 au tas que l'on a construit ci-dessus :



4.1 Présupposés

On suppose disposer de deux pointeurs

```
int *DebutTableau ;
int *FinTableau ;
```

le premier pointant au début d'un tableau d'entier et le second sur sa fin. On ne souhaite pas utiliser d'autre espace mémoire que ce tableau (notre tri est donc destructif). De plus, les tas binaires intermédiaires utilisent donc ce tableau comme espace de stockage.