

Algorithmique – Programmation Objet – Python  
TD n° 8  
Algorithmes de tri

Licence Informatique 2ème année  
Université de Nice Sophia Antipolis

## 1 Tri par Fusion

Soient  $T_1$  un tableau de  $n_1$  éléments triés par ordre croissant et  $T_2$  un tableau de  $n_2$  éléments triés par ordre croissant.

1. Écrivez un algorithme qui construit le tableau  $T$  constitué des éléments de  $T_1$  et de  $T_2$ , triés par ordre croissant.
2. Donnez la complexité de cet algorithme.
3. Considérez que vous avez un tableau de taille 16. Comment pouvez-vous appliquer le principe précédent pour obtenir un algorithme efficace de tri ? Écrivez le pseudo-code de cet algorithme. Généralisez cet algorithme à toute taille de tableau. Donnez sa complexité.

## 2 Structure de Tas

- Un tas descendant est un arbre binaire vérifiant les propriétés suivantes :
- la différence maximale de profondeur entre deux feuilles est de 1 (i.e. toutes les feuilles se trouvent sur la dernière ou sur l'avant-dernière ligne) ;
  - les feuilles de profondeur maximale sont "tassées" sur la gauche.
  - chaque nœud est de valeur inférieure à celle de ces deux fils.

Un tas ou un arbre binaire presque complet peut être stocké dans un tableau, en posant que les deux descendants de l'élément d'indice  $n$  sont les éléments d'indices  $2n$  et  $2n + 1$  (pour un tableau indicé à partir de 1). En d'autres termes, les nœuds de l'arbre sont placés dans le tableau ligne par ligne, chaque ligne étant décrite de gauche à droite.

L'insertion d'un élément dans un tas se fait de la façon suivante : on place l'élément sur la première case libre et on échange l'élément et son père quand ce dernier est supérieur et qu'il existe.

Dessinez un tas correspondant au tableau  $T : [2, 3, 8, 5, 1, 7, 4, 6]$

L'opération de tamisage consiste à échanger la racine avec le plus petit de ses fils, et ainsi de suite récursivement jusqu'à ce qu'elle soit à sa place. Remplacez la racine du tas par la valeur 7 et détaillez le tamisage.

Appliquez l'algorithme suivant pour trier le tableau  $T$  (on fera des dessins successifs montrant l'évolution du tas) :

- on commence par transformer le tableau en tas descendant ;
- on échange la racine avec le dernier élément du tableau, et on restreint le tas en ne touchant plus au dernier élément, c'est-à-dire à l'ancienne racine ;
- on tamise la racine dans le nouveau tas, et on répète l'opération sur le tas restreint jusqu'à l'avoir vidé et remplacé par un tableau trié.

Est-ce que cet algorithme trie le tableau  $T$  de façon croissante ou décroissante ?

### 3 Tri par base

1. Faites un tri par base (10) du tableau suivant 122, 12, 22, 342, 111, 731, 428, 513, 23, 5
2. Donnez les avantages et les inconvénients des tris par base suivant que l'on utilise la base 10, la base 2 et la base 256

### 4 Complexité des Algorithmes de Tri

Pour chacun des cas suivants donnez l'algorithme de tri qui sera certainement le plus efficace

1. on a beaucoup de nombre entre 1 et 100
2. on a  $n$  nombres entre 1 et  $n$
3. on a peu de nombres entre 1 et 1 milliard
4. on a très peu de nombres
5. on a beaucoup de nombres entre 1 et 1 milliard