

1. Et si on triait?

Dans cette exercice, nous allons revoir brièvement les 3 tris simples vu en cours:

(a) Tri par sélection

Une méthode de tri simple est le tri par sélection, il s'agit de chercher le plus petit élément du tableau $T[1..N]$ et on l'échange avec $T[1]$. Puis on recommence en cherchant le plus petit élément de $T[2..n]$ qu'on échange avec $T[2]$. Et ainsi de suite...

```
for (int i=1; i<=n-1; ++i) {
    trouver minimum A[j] de A[i]..A[n] Echanger A[i] et A[j]
}
```

(b) Tri par Bulle

Le tri par bulle n'est pas le tri, le plus efficace mais permet néanmoins de trier une liste avec une complexité $O(n^2)$.

```
for (int i=1; i<=n-1; ++i)
    for (int j=n; j>=i+1; --j)
        if (A[j]<A[j-1])
            Echanger A[j] et A[j-1];
```

(c) Tri par insertion L'idée de cet algorithme est de trier les éléments du tableau un par un, en insérant chaque élément à la bonne place dans la partie du tableau déjà triée. Pensez à la façon dont vous trieux un jeu de carte.

```
for (int i=2; i<=n; ++i)
    insérer A[i] a la bonne position dans A[1], ..., A[i-1]
```

Maintenant, prenons les listes suivantes:

(a) [4, 3, 2, 1]

(b) [2, 4, 1, 3]

1. Pour chacune de ces listes présenter le résultat de chaque itération de boucle for principale pour chacun des algorithmes

2. Heap

Illustrer l'opération Buildheap en la simulant sur le heap 4,1,3,2,16,9,10,14,8,7.

1. Dessiner l'arbre correspondant
2. Indiquer la suite des appels Heapify qui vont changer l'arbre et dessiner l'arbre résultant de chaque tel appel.

3. Priority Queue

Une "priority queue" (file à priorité) est une structure de données qui maintient un ensemble de données ayant une clé qui est un entier et supportant les opérations suivantes:

- $\text{Insert}(S, x)$: insere l'element x dans l'ensemble S
- $\text{Maximum}(S)$: retourne l'element avec la plus grande cle
- $\text{ExtractMax}(S)$: retourne et supprime l'element de S avec la plus grande cle.

(Les priority queues sont utilisees par exemple dans les systemes d'exploitation pour ranger les taches par priorite.)

Montrez qu'on peut implementer une priority queue avec un heap de facon a ce que l'operation $\text{Maximum}(S)$ prend un temps $O(1)$ et les autres operations s'executent en temps $O(\log n)$. Donner le pseudo-code pour chaque operation et faire une analyse.