

## TP Option Info MP/MP\* : Tri par tas

Dans ce TP, on travaille avec des tas-max, on représente un tas par un tableau de type `int array`, et on numérote les éléments à partir de 1. Par conséquent un tas à  $n - 1$  éléments est représenté par un tableau de taille  $n$  dans lequel la case 0 est inoccupée (on pourra y mettre un 0 par exemple). Les fonctions que l'on écrira modifient un tableau donné en entrée sans créer un nouveau tableau (elles ne renverront donc rien). Tester chaque fonction sur un exemple !

### Construction d'un tas

1. Écrire une fonction `echange t i j` prenant en entrée un tableau  $t$  et deux indices  $i$  et  $j$  et qui échange le contenu des cases d'indices  $i$  et  $j$ .
2. Écrire une fonction `montee t i` prenant en entrée un tableau  $t$  et un indice  $i$ , et qui fait remonter l'élément initialement en place  $i$  tant qu'il est supérieur à son père.
3. Écrire une fonction `descente t n i` prenant en entrée un tableau  $t$ , un indice  $n$  et un indice  $i < n$ , et qui fait descendre l'élément initialement en place  $i$  tant qu'il est inférieur à au moins un fils situé à un indice  $\leq n$  (attention : un noeud n'a pas toujours deux fils !). L'indice  $n$  indique la "fin" du tas à considérer, qui ne correspondra pas forcément à la taille du tableau (servira pour le tri par tas, où la taille du tas diminue au fur et à mesure).
4. En utilisant les fonctions précédentes, écrire une fonction `creer_tas_montee t` (resp. `creer_tas_descente t`) qui transforme le tableau  $t$  en tas en utilisant uniquement des montées (resp. des descentes). Attention à l'ordre de parcours du tableau !

Exemple : en partant du tableau  $t = [0; 1; 4; 5; 2; 7; 3]$ , on doit obtenir après l'appel `creer_tas_montee t` le tableau  $[0; 7; 5; 4; 1; 2; 3]$ . En repartant du tableau initial, on doit obtenir après l'appel `creer_tas_descente t` le tableau  $[0; 7; 4; 5; 2; 1; 3]$ .

### Tri par tas

Écrire la fonction `tri_tas t` qui réalise le tri par tas en place (sans créer de nouveau tableau) en temps  $O(n \log(n))$ . Attention, la taille du tas diminue à chaque étape !