

### Exercice 1 : Tri par sélection

Le principe du tri par sélection est le suivant :

- on cherche le plus petit élément de la liste,
- on le place en première position en effectuant un échange,
- on cherche le plus petit élément de liste à partir du deuxième élément,
- on le place en première position en effectuant un échange,
- on réitère le procédé.

Par exemple, pour  $l = [1, 5, 7, 4, 2, 6]$  :

- le plus petit élément de la liste est 1, il est déjà en première position mais on l'échange avec lui même, on obtient :  $l = [1, 5, 7, 4, 2, 6]$ ,
- le plus petit élément de la liste à partir du rang 2 est 2, on l'échange avec le deuxième élément, on obtient :  $l = [1, 2, 7, 4, 5, 6]$ ,
- le plus petit élément de la liste à partir du rang 3 est 4, on l'échange avec le troisième élément, on obtient :  $l = [1, 2, 4, 7, 5, 6]$ ,
- le plus petit élément de la liste à partir du rang 4 est 5, on l'échange avec le quatrième élément, on obtient :  $l = [1, 2, 4, 5, 7, 6]$ ,
- le plus petit élément de la liste à partir du rang 5 est 6, on l'échange avec le cinquième élément, on obtient :  $l = [1, 2, 4, 5, 6, 7]$ ,
- la liste est triée.

1. Ecrire une fonction `TriSelection` respectant ce principe.
2. Etudier les propriétés de la fonction `TriSelection`.

### Exercice 2 : Tri par insertion

Le principe du tri par insertion est le suivant :

- on suppose que les éléments indexés de 0 à  $i-1$  sont triés,
- on considère  $l[i]$  comme étant une clé que l'on va insérer parmi les éléments déjà triés,
- on compare la clé aux éléments déjà triés en partant du plus grand,
- on décale d'un rang vers la droite les éléments plus grand que la clé,
- on place la clé lorsqu'on arrive à un élément plus petit que la clé.

Par exemple, pour  $l = [1, 5, 7, 4, 2, 6]$  et  $i = 3$ , la clé est  $l[i] = 4$  et :

- on décale 7 :  $[1, 5, 7, 7, 2, 6]$ ,
- on décale 5 :  $[1, 5, 5, 7, 2, 6]$ ,
- on ne décale pas 1 donc on insère la clé :  $[1, 4, 5, 7, 2, 6]$ .

1. Ecrire une fonction `TriInsertion` respectant ce principe.
2. Etudier les propriétés de la fonction `TriInsertion`.

### Exercice 3 : Tri par comptage

Le tri par comptage ne s'applique qu'aux listes d'entiers naturels dont on connaît un majorant. On considèrera une telle liste  $l$  et un majorant  $m$ .

1. Ecrire une fonction `comptage` qui prend comme argument  $l$  et  $m$  et qui renvoie une liste  $L$  de longueur  $m + 1$  tel que, pour tout  $k \in \llbracket 0, m \rrbracket$ ,  $L[k]$  soit égal au nombre d'occurrences de  $k$  dans  $l$ .
2. Le principe du tri par comptage est parcourir la liste  $L$  afin de rajouter les éléments qui sont dans  $l$  autant de fois qu'ils apparaissent dans  $l$ .  
Ecrire une fonction `TriComptage` qui effectue un tri par comptage.
3. Etudier les propriétés de la fonction `TriComptage`.

#### Exercice 4 : Tri cocktail

Le principe du tri cocktail est le principe du tri à bulles mais effectué dans les deux sens :

- on fait remonter en dernière position l'élément le plus grand,
- on fait redescendre en première position l'élément le plus petit,
- on recommence en ne considérant plus les deux éléments déjà triés.

Par exemple, pour  $l = [8, 6, 5, 3, 2]$  :

- on fait remonter 8 :  $l = [6, 5, 3, 2, 8]$ ,

- on fait redescendre 2 :  $l = [2, 6, 5, 3, 8]$ ,
- on ne travaille plus que sur la sous-liste  $[6, 5, 3]$ ,
- on fait remonter 6 :  $l = [2, 5, 3, 6, 8]$ ,
- on fait redescendre 3 :  $l = [2, 3, 5, 6, 8]$ ,
- la liste est triée.

1. Ecrire une fonction `TriCocktail` respectant ce principe.
2. Etudier les propriétés de la fonction `TriCocktail`.