Lycée Marcelin BERTHELOT
BCPST 1C – Mme MOREL

Tri par sélection

1 Principe

Le **tri par sélection** consiste à construire au fur et à mesure la liste triée en sélectionnant tout d'abord le plus petit élément, puis le deuxième plus petit, puis le troisième, etc.
On part donc d'une liste de nombres quelconque, et en lui appliquant le tri par sélection, on obtient une liste de nombres rangés dans l'ordre croissant.
Afin de procéder à ce tri, on a besoin d'une fonction `ind_min` renvoyant **l'indice** du plus petit élément d'une liste de nombres `L`:

```
def ind_min(L):
```

2 Première version non optimale

Une première version de cet algorithme réalisant le tri d'une liste `L` en une liste `L_triee` consiste à :

- partir de la liste vide `L_triee = [ ]`
- trouver le minimum de `L`, le placer dans `L_triee` et l'enlever de `L`
- trouver le deuxième plus petit élément de la liste de départ : c'est en fait le nouveau minimum de `L`, le placer dans `L_triee` et l'enlever de `L`
- continuer jusqu'à avoir traité tous les éléments de `L`

Exemple 1 On considère la liste `L=[4,7,2,5,1,8,6,3]`.
Compléter le tableau suivant indiquant ce que contiennent successivement les listes `L` et `L_triee` à chaque étape de cet algorithme :

étapes de l'algo	`L`	`L_triee`
0	$[4,7,2,5,1,8,6,3]$	$[\,]$
1	$[4,7,2,5,8,6,3]$	
2	$[4,7,5,8,6,3]$	$[1,2]$
3		$[1,2,3]$
4	$[7,5,8,6]$	$[1,2,3,4]$
5	$[7,8,6]$	
6		
7		
8	$[\,]$	$[1,2,3,4,5,6,7,8]$

Algorithme 1 Compléter la fonction `tri_select_1` prenant en argument une liste de nombres `L` et renvoyant la liste triée `L_triee`:

```
def tri_select_1 (L):
    L_triee = ....
    for etape in range(    ):
        k=ind_min(L)
        ....
        .....
    return   ....
```

Remarque 1 Les comparaisons d'éléments ont lieu lorsqu'on utilise la fonction `ind_min` : obtenir `ind_min(L)` où `L` est une liste comportant m éléments nécessite donc m opérations élémentaires (i.e. ici de comparaisons entre deux éléments de la liste).
Dans la fonction `tri_select_1`, on appelle `ind_min(L)` à chaque étape de la boucle "`for etape in range( len(L) )`" sur une liste `L` comptant (`len(L)` − `etape`) éléments.

Si on note n =`len(L)`, on a donc en tout $n + (n-1) + (n-2) + \cdots + 1 = \dfrac{n(n-1)}{2}$ opérations élémentaires pour trier une liste de longueur n.

Remarque 2 Cette façon de faire n'est pas satisfaisante du point de vue de l'espace mémoire sollicité (on utilise $2n$ emplacements pour traiter une liste de taille n puisqu'on utilise deux listes `L` et `L_triee`). On dit que la version `tri_select_1` du tri par sélection n'est pas *en place*.

3 Deuxième version "en place"

Dans la version *en place* du tri par sélection, on travaille par modifications de la liste `L` uniquement, plus précisément par échanges d'éléments.
On a donc besoin d'une fonction `echange(L,p,q)` permettant d'échanger deux éléments d'une liste `L` donnés par leurs indices `p` et `q`:

```
def echange(L,p,q):
```

Une deuxième version du tri par sélection consiste donc à :

- trouver le minimum de la liste `L`
- l'échanger avec le premier élément de `L` c'est-à-dire avec `L[0]`
- trouver le deuxième plus petit élément de `L` : c'est en fait le minimum de la sous-liste de `L` ne contenant pas le premier élément c'est-à-dire de `L[1:]`
- l'échanger avec le deuxième élément de `L` c'est-à-dire avec `L[1]`
- continuer jusqu'à avoir traité tous les éléments de `L`

Exemple 2 On considère la liste `L=[6, 3, 1, 5, 4, 8, 7, 2]`.
Compléter le tableau suivant indiquant ce que contient successivement la liste `L` à chaque étape de cet algorithme :

étapes de l'algo	`L`
0	$[6,3,1,5,4,8,7,2]$
1	$[1,3,6,5,4,8,7,2]$
2	$[1,2,6,5,4,8,7,3]$
3	
4	$[1,2,3,4,5,8,7,6]$
5	
6	$[1,2,3,4,5,6,7,8]$
7	
8	

Algorithme 2 :

- on travaille sur la sous-liste `[L[etape],L[etape+1],...,L[len(L)-1]]` car les éléments précédents sont déjà triés,
- Attention ! l'indice `k = ind_min(sous_liste)` est l'indice *dans la sous-liste* : ce n'est pas le même que l'indice dans la liste `L`,

- on échange le minimum de la sous-liste (qui est d'indice `k+etape` dans `L`) avec la valeur courante.

Compléter la fonction `tri_select_2` prenant en argument une liste de nombres `L` et la triant par sélection, renvoyant ainsi la liste triée `L`:

```
def tri_selection_2(L):
    for etape in range (      ) :
        sous_liste = L[         :            ]
        k=ind_min(            )
        ......
    return ......
```