

Tri par sélection

1 Algorithme

Le tri par sélection recherche le plus petit ou le plus grand élément non trié de la liste pour le mettre à son emplacement définitif, qui peut varier si l'on souhaite trier en ordre croissant ou décroissant.

On répète cette recherche autant de fois qu'il y a d'éléments dans la liste. Chaque nouvelle itération s'effectue donc sur une liste plus petite qu'à l'itération précédente, car on ne considère plus les éléments déjà triés.

C'est un tri de complexité $O(n^2)$.

2 Exemple

Voici les étapes du tri par sélection sur la liste [6, 5, 3, 1, 8, 7, 2, 4].

On souhaite trier la liste en ordre croissant et on recherche les minimums pour les placer au début.
Les cases grisées sont les indices déjà triés.

indices	0	1	2	3	4	5	6	7
Valeurs	6	5	3	1	8	7	2	4

indices	0	1	2	3	4	5	6	7
Iter 0	1	5	3	6	8	7	2	4

indices	0	1	2	3	4	5	6	7
Iter 1	1	2	3	6	8	7	5	4

indices	0	1	2	3	4	5	6	7
Iter 2	1	2	3	6	8	7	5	4

indices	0	1	2	3	4	5	6	7
Iter 3	1	2	3	4	8	7	5	6

indices	0	1	2	3	4	5	6	7
Iter 4	1	2	3	4	5	7	8	6

indices	0	1	2	3	4	5	6	7
Iter 5	1	2	3	4	5	6	8	7

indices	0	1	2	3	4	5	6	7
Iter 6	1	2	3	4	5	6	7	8

Indices parcourus : de 0 à 7

Indice du minimum : 3

Indices des valeurs échangées : 0 et 3.

Indices parcourus : de 1 à 7

Indice du minimum : 6

Indices des valeurs échangées : 1 et 6.

Indices parcourus : de 2 à 7

Indice du minimum : 2

Indices des valeurs échangées : 2 et 2.

Indices parcourus : de 3 à 7

Indice du minimum : 7

Indices des valeurs échangées : 3 et 7.

Indices parcourus : de 4 à 7

Indice du minimum : 6

Indices des valeurs échangées : 4 et 6.

Indices parcourus : de 5 à 7

Indice du minimum : 7

Indices des valeurs échangées : 5 et 7.

Indices parcourus : de 6 à 7

Indice du minimum : 7

Indices des valeurs échangées : 6 et 7.

Il ne reste plus qu'une valeur à trier.

Elle est forcément à sa place donc la liste est triée.

3 Pseudo-code

```
procédure tri_selection(liste l)
    n ← longueur(l)
    pour i de 0 à n - 2
        min ← i
        pour j de i + 1 à n - 1
            si l[j] < l[min], alors min ← j
        fin pour
        si min ≠ i, alors échanger l[i] et l[min]
    fin pour
fin procédure
```

Exercice 1 : Trier dans l'ordre croissant la liste [1, 5, 3, 6, 8, 7, 2, 4] en utilisant l'algorithme de tri par sélection qui recherche les minimums.

indices	0	1	2	3	4	5	6	7
Valeurs	1	5	3	6	8	7	2	4
Iter 0								
Iter 1								
Iter 2								
Iter 3								
Iter 4								
Iter 5								
Iter 6								

Exercice 2 : Coder l'algorithme du tri par sélection en ordre croissant recherchant les minimums en Python. Aider vous du pseudo-code si besoin. Coder le test pour la liste de l'exercice 1.

```
1  def echange(l : list, i : int, j : int):
2      """ Échange les éléments d'indice i et j dans la liste l. """
3      temp = l[i]
4      l[i] = l[j]
5      l[j] = temp
6
7  def tri_par_selection(l : list):
8      """ Tri la liste l avec un tri par sélection
9      >>> tri_par_selection([1, 5, 3, 6, 8, 7, 2, 4])
10     [1, 2, 3, 4, 5, 6, 7, 8]
11     """
12
13  for i in range(len(l) - 1):
14      indice_min = i
15      for j in range(i, len(l)):
16          if l[indice_min] > l[j]:
17              indice_min = j
18      echange(l, i, indice_min) # ou l.insert(i, l.pop(indice_min))
19  return l
```

Bonus : Coder l'algorithme du tri par sélection en ordre croissant recherchant les maximums en Python.

Il y a plusieurs bonnes façons de coder les boucles.

Solution 1 :

```
1  def tri_par_selection_max_croissant(l : list):
2      """ Tri la liste l avec un tri par sélection
3      >>> tri_par_selection_max_croissant([1, 5, 3, 6, 8, 7, 2, 4])
4          [1, 2, 3, 4, 5, 6, 7, 8]
5
6
7      for i in range(len(l) - 1):
8          indice_max = 0
9          for j in range(0, len(l)- i):
10             if l[indice_max] < l[j]:
11                 indice_max = j
12             l.insert(len(l) - 1 - i, l.pop(indice_max))
13     return l
```

Solution 2 :

```
1  def tri_par_selection_max_croissant(l : list):
2      """ Tri la liste l avec un tri par sélection
3      >>> tri_par_selection_max_croissant([1, 5, 3, 6, 8, 7, 2, 4])
4          [1, 2, 3, 4, 5, 6, 7, 8]
5
6
7      for i in range(len(l), 0, -1):
8          indice_max = 0
9          for j in range(i):
10             if l[indice_max] < l[j]:
11                 indice_max = j
12             l.insert(i-1, l.pop(indice_max))
13     return l
```

N'hésitez pas à aller sur <https://pythontutor.com/> pour tester et visualiser votre code.