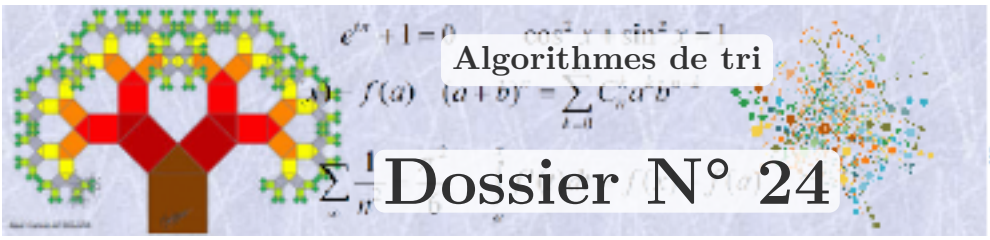


Travaux dirigés

Introduction

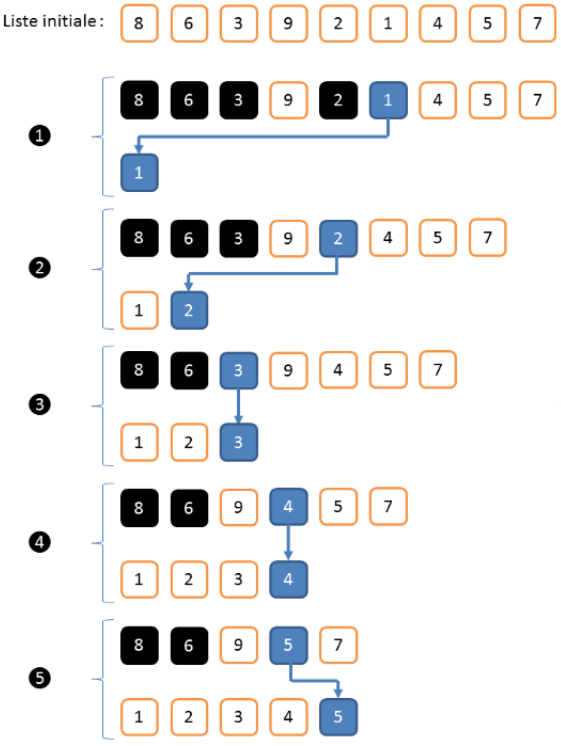
L'objectif de ce dossier est de construire des algorithme permettant de trier les valeurs d'une liste.
Attention, en informatique, on parle de tri mais il s'agit plutôt d'un classement dans un certain ordre.
Par exemple dans la vie courante, on trie les déchets mais on classe les élèves dans l'ordre alphabétique

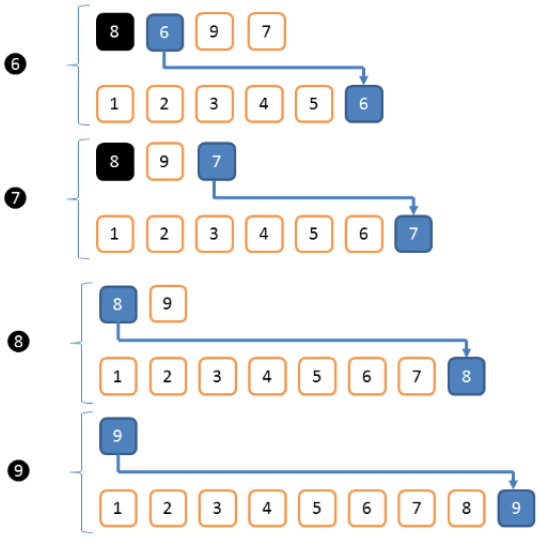
Exercice 1 :

1 Le tri par sélection d'une liste consiste à :

- Chercher la plus petite valeur de la liste
- La stocker en première place d'une liste chargée de donner la liste triée
- Recommencer avec la liste donnée au départ à laquelle on a retirée cette plus petite valeur

Voici un exemple





2 Faire de même avec la liste : $L = [5, 12, 3, 7, 4]$

3 Combien de comparaison ont été nécessaires pour trier, avec cette méthode, la liste $[8, 6, 3, 9, 2, 1, 4, 5, 7]$

Exercice 2 : valeur minimale d'une liste

1 La fonction python ci-dessous renvoie la plus petite valeur de la liste L donnée en paramètre :

```
def min(L):
  min=L[0]
  for j in range(1,len(L)):
    if L[j]<min:
      ----------
  return min
```

Compléter la partie manquante (en pointillé)

2 Écrire cette fois une fonction qui renvoie la plus petite valeur d'une liste ainsi que la position de cette valeur dans la liste de départ (son indice dans la liste en fait)

Exercice 3 : L'algorithme du tri pas sélection

1 Le programme python ci-dessous trie la liste L donnée en début de programme. Compléter les parties manquantes de l'algorithme
On utilisera, à bon escient les fonctions append() et remove()

```
L=[ 8, 6, 3, 9, 2, 1, 4, 5, 7 ]
tri=[]
while len(L)>=1:
  min=L[0]
  for i in range(1,len(L)):
    if L[i]<min:
      min=L[i]
  tri.-----------
  L.-----------
print(tri)
```

Exercice 4 : Une autre méthode de tri

Voici un programme python permettant de trie rune liste :

```
L=[ 8, 6, 3, 9, 2]
n=len(L)
for i in range(n-1):
  min=L[i]
  for j in range(i+1,n):
    if L[j]<min:
      min=L[j]
      L[i],L[j]=L[j],L[i]
print(L)
```

Mettre en oeuvre ce programme en écrivant la liste L à chaque étape de l'algorithme

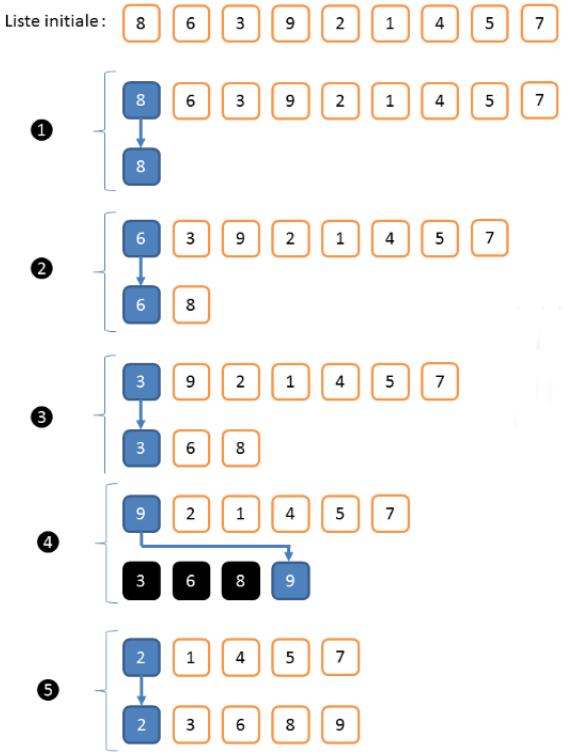
Remarque

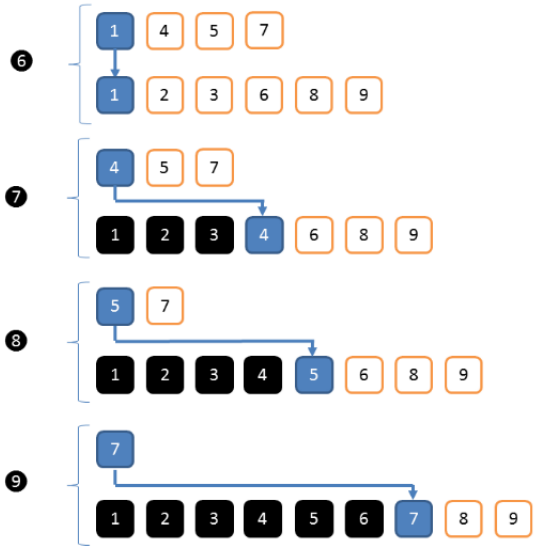
En fait on a même pas besoin du candidat au minimum :

```
L=[ 8, 6, 3, 9, 2, 1, 4, 5, 7 ]
n=len(L)
for i in range(n-1):
  for j in range(i+1,n):
    if L[j]<L[i]:
      L[i],L[j]=L[j],L[i]
print(L)
```

Exercice 5 : Tri par insertion

1 Le tri par insertion d'une liste est présenté ci-dessous





2 Faire de même avec la liste : $L = [5, 12, 3, 7, 4]$

3 Combien de comparaison ont été nécessaires pour trier, avec cette méthode, la liste $[8, 6, 3, 9, 2, 1, 4, 5, 7]$?

Information

le tri par insertion est basé sur le schéma algorithmique suivant :

```
FONCTION TriInsertion(L)
    POUR i VARIANT DE 2 A LONGUEUR DE L FAIRE
        INSERER L[i] à sa place dans L[1:i-1];
FIN FONCTION
```

Exercice 6 :

Voici l'algorithme du tri par insertion :

```
def tri_insertion(L):
    for i in range(1,len(L)):
        encours = L[i]
        j = i
        while j>0 and L[j-1]>encours:
            L[j]=L[j-1]
            j = j-1
        L[j]=encours


tab = [98, 22, 15, 32, 2, 74, 63, 70]

tri_insertion(tab)
```

Tester ce programme dans python tutor qui permet de visualiser pas à pas l'évolution du programme
http://www.pythontutor.com/visualize.html#mode=edit

Complément

Voici une vidéo bilan sur le tri par insertion à consulter :
lumni.fr/video/les-algorithmes-de-tri
En voici une autre :
https://interstices.info/les-algorithmes-de-tri(triparinsertion)
Et celle ci : https://www.podcastscience.fm/dossiers/2014/09/04/les-tris/