



LES FACULTÉS
DE L'UNIVERSITÉ
CATHOLIQUE DE LILLE

TP N°7 :

Programmation en Python

Les tris





SOMMAIRE

Présentation du document	1
Sommaire	2
I. Tri par insertion	3
1. Vidéo	3
2. Principe de fonctionnement	3
3. Voici l'algorithme du tri par insertion :	4
4. Exercices sur feuille	4
5. Exercices sur ordinateur	4
II. Tri par selection	4
1. Vidéo	4
2. Principe de fonctionnement	5
3. Voici l'algorithme	5
4. Exercice sur feuille	5
III. Extension : recherche dichotomique	6
1. Principe	6
2. Fonctionnement	6
3. Résumé	6
4. Algorithme	7
5. Exercices sur feuille	7
6. Sur ordinateur	8

Le terme *tri* désigne l'opération consistant à ordonner un ensemble d'éléments en fonction de clés sur lesquelles est définie une relation d'ordre.

Tâche N°1. Pour commencer je vous invite à vous rendre sur le site <http://lwh.free.fr/pages/algo/tri/tri.htm> pour essayer de trier des tonneaux.

Commencer par sélectionner le nombre de tonneaux, avec la souris déplacer les tonneaux sur la balance ou sur le zone de stockage jusqu'à ce que vous pensiez les avoir trier.

Appuyez alors sur terminer pour vérifier que les tonneaux sont correctement trié.

Le site mesure le nombre de comparaison que vous faites ce qui est un bon indicateur de la bonne complexité de votre stratégie.

Le but du jeu est donc de réussir à les tris avec le moins de comparaison possible, essayez de mettre en place plusieurs stratégies

<ul style="list-style-type: none">• Commencez par trois tonneaux• Cliquez sur <i>Mélanger</i>• Et quand vous avez fini cliquez <i>Terminé !</i> pour vérifier <p>Ensuite ré-essayez en augmentant le nombre de tonneaux</p>	<p>Comparaisons : <input type="text" value="0"/></p> <p>Déplacements : <input type="text" value="0"/></p> <p>Nombre de tonneaux : <input type="text" value="3"/></p> <p><input type="button" value="Mélanger"/></p> <p><input type="button" value="Terminé !"/></p> <p><input type="button" value="Résoudre"/></p>
---	--

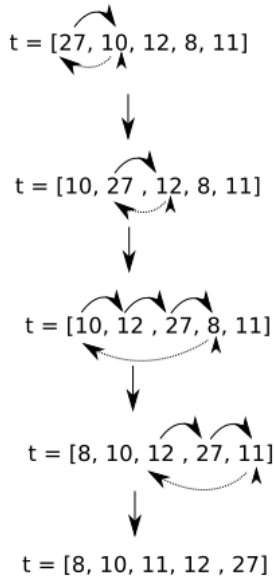
I. TRI PAR INSERTION

1. Vidéo

Lisez : <https://www.youtube.com/watch?v=bRPHvWgc6YM>

2. Principe de fonctionnement

On peut résumer le principe de fonctionnement de l'algorithme de tri par insertion avec le schéma suivant :



3. Voici l'algorithme du tri par insertion :

```
VARIABLE
t : tableau d'entiers
i : nombre entier
j : nombre entier
k : nombre entier
DEBUT
j←2
tant que j<=longueur(t) :
  i←j-1
  k←t[j]
  tant que i>0 et que t[i]>k:
    t[i+1]←t[i]
    i←i-1
  fin tant que
  t[i+1]←k
  j←j+1
fin tant que
FIN
```

4. Exercices sur feuille

Tâche N°2. Produisez un schéma équivalent à celui du principe de fonctionnement pour le tableau [12, 8, 23, 10, 15]

Tâche N°3. Appliquez cet algorithme au tableau t = [27, 10, 12, 8, 11] et vérifiez que vous obtenez bien le tableau [8, 10, 11, 12, 27]

5. Exercices sur ordinateur

Tâche N°4. Traduisez cet algorithme en langage Python

II. TRI PAR SELECTION

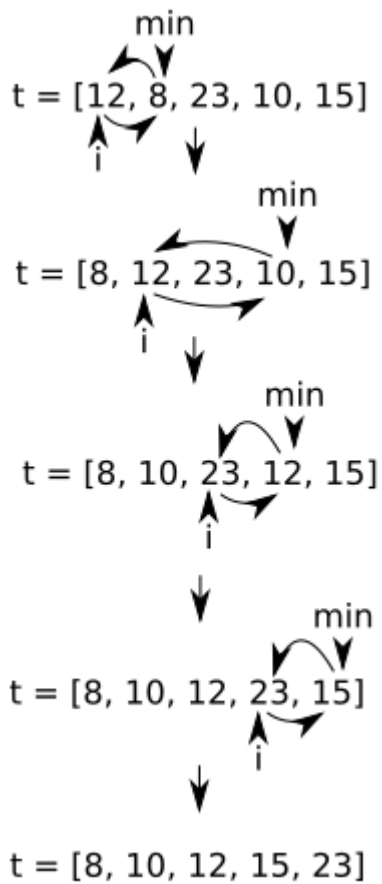
1. Vidéo

<https://www.youtube.com/watch?v=8u3Yq-5DTN8>



2. Principe de fonctionnement

On peut résumer le principe de fonctionnement de l'algorithme de tri par sélection avec le schéma suivant :



3. Voici l'algorithme

```
VARIABLE
t : tableau d'entiers
i : nombre entier
min : nombre entier
j : nombre entier
DEBUT
i←1
tant que i<longueur(t): //boucle 1
  j←i+1
  min←i
  tant que j<=longueur(t): //boucle 2
    si t[j]<t[min]:
      min←j
    fin si
    j←j+1
  fin tant que
  si min≠i :
    échanger t[i] et t[min]
  fin si
  i←i+1
fin tant que
FIN
```

4. Exercice sur feuille

Tâche N°5. Produisez un schéma équivalent à celui du principe de fonctionnement pour le tableau



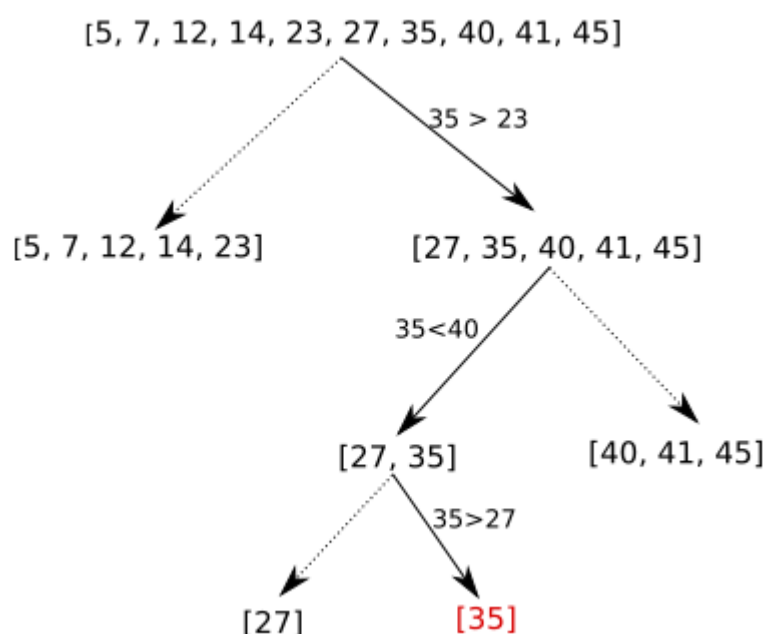
[15, 16, 11, 13, 12]

Tâche N°6. Appliquez cet algorithme au tableau $t = [27, 10, 12, 8, 11]$ et vérifiez que vous obtenez bien le tableau $[8, 10, 11, 12, 27]$

III. EXTENSION : RECHERCHE DICHOTOMIQUE

1. Principe

Il est aussi possible de représenter le principe de l'algorithme de recherche dichotomique avec le schéma suivant (on recherche la valeur $x=35$ dans le tableau $t = [5, 7, 12, 14, 23, 27, 35, 40, 41, 45]$) :

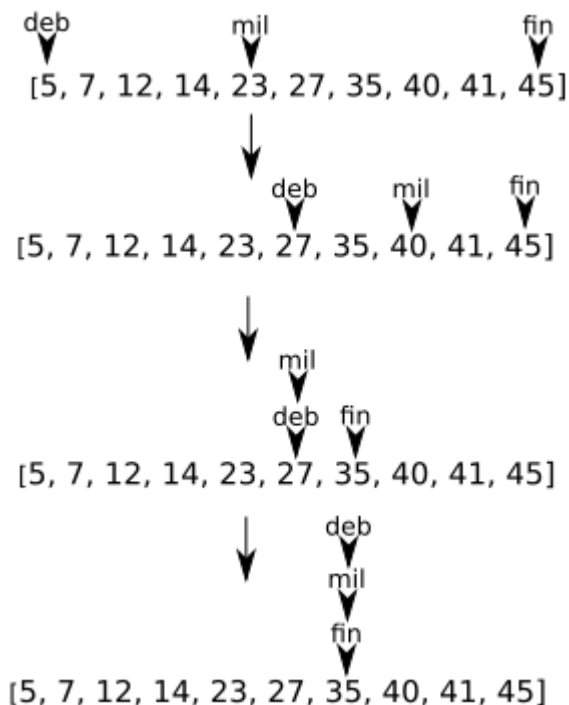


2. Fonctionnement

Dans le schéma ci-dessus, à chaque étape, on garde uniquement le tableau désigné par la flèche en trait plein, on abandonne le tableau désigné par la flèche en pointillé. Dans la première étape on part de tableau [5, 7, 12, 14, 23, 27, 35, 40, 41, 45], ce tableau est divisé en 2 tableaux : [5, 7, 12, 14, 23] et [27, 35, 40, 41, 45]. La valeur recherchée (35) ne peut pas être dans le premier tableau ([5, 7, 12, 14, 23]) puisque $35 > 23$ et que les autres valeurs du premier tableau sont forcément plus petite que 23 (le tableau est trié). On garde donc uniquement le second tableau ([27, 35, 40, 41, 45]) et on recommence le processus (on divise ce tableau en deux...) jusqu'au moment où l'on "tombe" sur la valeur recherchée ou que l'on se retrouve avec un tableau contenant un seul élément : si l'élément unique du tableau n'est pas l'élément recherché, cela signifie que l'élément recherché n'est pas dans le tableau.

3. Résumé

On peut résumer le principe de fonctionnement de l'algorithme de recherche dichotomique par le schéma suivant :



Les variables *deb*, *mil* et *fin* du schéma correspondent aux variables *deb*, *mil* et *fin* de l'algorithme.

4. Algorithme

```
VARIABLE  
t : tableau d'entiers trié  
mil : nombre entier  
fin : nombre entier  
deb : nombre entier  
x : nombre entier // x : l'entier recherché  
tr : booléen  
DEBUT  
tr ← FAUX  
deb ← 1  
fin ← longueur(t)  
tant que tr == FAUX et que deb ≤ fin :  
  mil ← partie_entière((deb+fin)/2)  
  si t[mil] == x :  
    tr = VRAI  
  sinon :  
    si x > t[mil] :  
      deb ← mil+1  
    sinon :  
      fin ← mil-1  
  fin si  
fin tant que  
renvoyer la valeur de tr  
FIN
```

5. Exercices sur feuille

Tâche N°7. Faites un schéma permettant d'expliquer le principe de la recherche dichotomique en vous basant sur l'exemple suivant : $t = [5, 7, 12, 14, 23, 27, 35, 40, 41, 45]$ et $x = 40$ (valeur recherchée)

Tâche N°8. Faites un schéma permettant d'expliquer le principe de la recherche dichotomique en vous



basant sur l'exemple suivant : $t = [5, 7, 12, 14, 23, 27, 35, 40, 41, 45]$ et $x = 9$ (valeur recherchée)

Tâche N°9. Appliquez cet algorithme au tableau $t = [5, 7, 12, 14, 23, 27, 35, 40, 41, 45]$ et $x = 40$.
Vérifiez que l'algorithme renvoie bien VRAI

Tâche N°10. 2) Appliquez cet algorithme au tableau $t = [5, 7, 12, 14, 23, 27, 35, 40, 41, 45]$ et $x = 9$. Vérifiez que l'algorithme renvoie bien FAUX

6. Sur ordinateur

Tâche N°11. Implémentez l'algorithme de recherche dichotomique en Python.

