Les méthodes de tri et de recherche

A. Présentation des algorithmes de tri

Un algorithme de tri est une suite finie d'instructions servant à réordonner **une séquence d'éléments** suivant un critère fixé à priori.

Il existe plusieurs méthodes de tri, on cite par exemple :

- **La méthode de tri par sélection**
- **La méthode de tri à bulles**

Remarque : Dans tout ce qui suit, on suppose que l'on trie **des nombres entiers** par **ordre croissant** et que ceux-ci se trouvent dans **un tableau T**.

B. Les Méthodes de tri

1. Tri par sélection

o Principe

L'algorithme de ***tri par sélection*** consiste à :

1- Trouver la position **posmin** de l'élément le plus **petit** du tableau

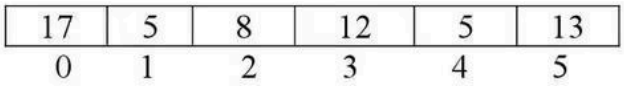
2- Une fois cette position **posmin** trouvée, on compare les éléments **T[0]** et **T[posmin],** s'ils sont différents on fait leur permutation.

3- On recommence ces opérations sur la suite **(T[1], T[2], …, T[N-1])**, ainsi on recherche le plus petit élément de cette nouvelle suite et on l'échange avec **T[1]**. Et

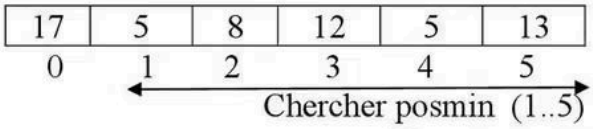
ainsi de suite jusqu'au moment où on n'a plus qu'une suite composée d'un seul élément **T[N-1]**.

- **Exemple n=6**

T

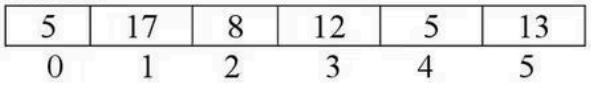


➢ **i=0, posmin=0**

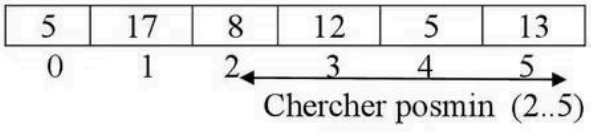


Posmin=1

T[0] ≠ T[1] Permutation



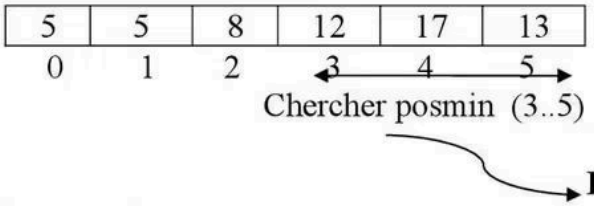
➢ **i=1, posmin=1**



Posmin=4

T[1] ≠ T[4] Permuter

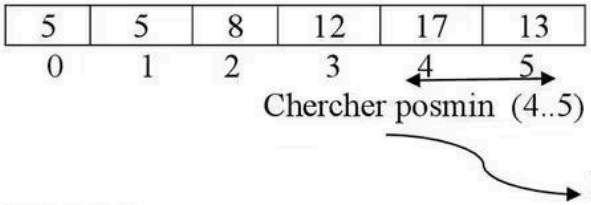
➢ **i=2, posmin=2**



Posmin=2

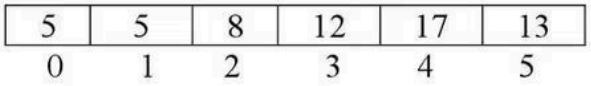
T[2] = T[2] Pas de Permutation

➢ **i=3, posmin=3**

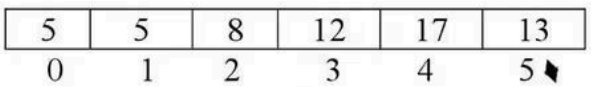


Posmin=3

T[3] =T[3] Pas de Permutation



➢ **i=4, posmin=4**



Chercher posmin (5) **posmin=5**

T[4] ≠ T[5] Permutation

5	5	8	12	13	17
0	1	2	3	4	5

Procédure tri_selection (n:entier, @ T:tab) **TDOL**
Debut

```
Pour i de 0 à n-2   faire
  posmin← i
  pour j de i+1 à n-1 faire
      si T[j] < T[posmin] alors
          posmin ← j
      fin si
  fin pour
  Si (T[posmin] ≠ T[i]) Alors
          aux ← T[posmin]
          T[posmin] ← T[i]
           T[i] ← aux
  Finsi
Fin Pour
```

Fin

O	T/N
i	Entier
j	Entier
posmin	Entier
aux	Entier

2. Tri à bulles

o Principe :

L'algorithme du **tri à bulles** consiste à comparer les différentes valeurs consécutives du tableau **T** et à les permuter s'ils ne sont pas dans le bon ordre.

1. **Comparer** les deux premiers éléments du tableau, si le premier est supérieur au second, **une permutation** est effectuée.
2. Ensuite, sont **comparées et** éventuellement **permutées** les valeurs d'indices **1** et **2**, **2** et **3** jusqu'à **(n-2)** et **(n-1)**
3. Une fois cette étape achevée, il est certain que **le dernier élément** du tableau est **le plus grand**. L'algorithme reprend donc pour classer les (n-1) éléments qui précèdent.
4. L'algorithme se termine quand **il n'y a plus de permutations possibles**.

o Exemple n=6

Etape1 T

17	5	8	12	5	13
0	1	2	3	4	5

➢ **I=0,Permuter=faux**

17	5	8	12	5	13
0	1	2	3	4	5

T[0] > T[1] donc Permutation et Permuter=vrai

5	17	8	12	5	13
0	1	2	3	4	5

- **i=1,Permuter=vrai**

T[1] > T[2] donc Permutation et Permuter=vrai

5	8	17	12	5	13
0	1	2	3	4	5

- **i=2,Permuter=vrai**

T[2] > T[3] donc Permutation et Permuter=vrai

5	8	12	17	5	13
0	1	2	3	4	5

- **i=3,Permuter=vrai**

T[3] > T[4] donc Permutation et Permuter=vrai

5	8	12	5	17	13
0	1	2	3	4	5

- **i=4,Permuter=vrai**

T[4] > T[5] donc Permutation et Permuter=vrai

5	8	12	5	13	17
0	1	2	3	4	5

Remarque : Après le premier parcours, le plus grand élément étant à sa position définitive, il n'a plus à être traité.

Etape2

- **i=0,Permuter=faux**

T[0] < T[1] donc pas de Permutation

5	8	12	5	13	17
0	1	2	3	4	5

- **i=1,Permuter=faux**

T[1] < T[2] donc pas de Permutation

5	8	12	5	13	17
0	1	2	3	4	5

- **i=2,Permuter=faux**

T[2] > T[3] donc Permutation ,permuter=vrai

5	8	5	12	13	17
0	1	2	3	4	5

- **i=3,Permuter=vrai**

T[3] < T[4] donc pas de Permutation

5	8	5	12	13	17
0	1	2	3	4	5

- **i=4,Permuter=vrai**

T[4] < T[5] donc pas de Permutation

5	8	5	12	13	17
0	1	2	3	4	5

Etape3

- **i=0,Permuter=faux**

T[0] < T[1] donc Pas de Permutation

5	8	5	12	13	17
0	1	2	3	4	5

- **i=1,Permuter=faux**

T[1] > T[2] donc Permutation ,permuter=vrai

5	5	8	12	13	17
0	1	2	3	4	5

- **i=2,Permuter=vrai**

T[2] < T[3] donc Pas de Permutation

- **i=3,Permuter=vrai**

T[3] < T[4] donc Pas de Permutation

- **i=4,Permuter=vrai**

T[4] < T[5] donc Pas de Permutation

Etape4

- **i=0,Permuter=faux**

T[0] = T[1] donc Pas de Permutation

5	5	8	12	13	17
0	1	2	3	4	5

- **i=1,2 ,3,4 Permuter=faux on sort avec un tableau trié**

- **Algorithme de la procédure Tri_bulles dans un tableau T**

Procedure tri_bulles (n:entier, @ T : Tab) **TDOL**

Début

répéter

permuter ← faux

```
    Pour i de 0 à n -2  faire
        si (T[i] > T[i+1]) alors
              aux ← T[i]
              T[i] ← T[i+1]
              T[i+1] ← aux
              permuter ← vrai
        Finsi
   Fin pour
  Jusqu'à (permuter = faux)
Fin
```

O	T/N
permuter	Booleen
i	Entier
aux	Entier

C. Les algorithmes de recherche

Un algorithme de Rechercher revient à vérifier l'existence d'un élément **E** dans un ensemble. Il existe deux méthodes de recherche :

- **La méthode de recherche séquentielle**
- **La méthode de recherche dichotomique**

1. La méthode de recherche séquentielle

o Principe

La méthode de **recherche séquentielle** d'un élément dans un ensemble consiste à parcourir l'ensemble **élément** par **élément** en les **comparant** avec **l'élément à chercher** jusqu'à **trouver** ce dernier ou **achever** l'ensemble.

o Algorithme de la fonction recherche_sequentielle d'un entier x dans un tableau t (méthode 1)

Fonction recherche_sequentielle (x : entier ; n : entier ; t : tab) : booléen
Début
trouve ← faux
i ← 0
répéter
si t[i] = x alors
trouve← vrai
sinon
i ← i+1
finsi
jusqu'à (trouve=vrai) ou (i=n)
retourner trouve
Fin

Fonction recherche_dichotomique (x :entier ;n :entier;t:tab) :booléen
Debut
trouve ← faux
i ← 0
d ←0
f←n-1
répéter
milieu←(d+f) div 2
Si x= t[milieu] alors
trouve ←vrai
Sinon si x < t[milieu] alors
f ←milieu -1
Sinon
d ←milieu +1
Finsi
jusqu'à (trouve=vrai) ou (i=n)
Retourner trouve
Fin

TDOL

O	T/N
trouve	Booléen
i	Entier

- **Algorithme de la fonction recherche séquentielle d'un entier x dans un tableau t (méthode 2)**

Fonction recherche_sequentielle (x : entier ; n : entier ; t : tab) : booléen
Début
trouve ← faux
i ← 0
répéter
trouve← t[i] = x
i ← i+1
jusqu'à (trouve=vrai) ou (i=n)
retourner trouve
Fin

TDOL

O	T/N
trouve	Booléen
i	Entier

Activité3

Ecrire l'algorithme principal d'un programme qui permet de chercher un entier x pair saisie par l'utilisateur dans un tableau

- **Algorithme de la procédure tri_selection dans un tableau T**