

Les algorithmes de tri et de recherche

Définition :

Un algorithme de tri est une suite finie d'instructions servant à réordonner une séquence d'éléments, dans un ordre croissant ou décroissant, fixé à priori.

Les méthodes de tri :

A. Tri par sélection :

La méthode de tri par sélection du tableau T consiste à :

- 1- Commencer par $i = 1$ et chercher la position **pmin** du plus petit élément de T.
si $T[i] \neq T[pmin]$ alors on les permute. T[1] est maintenant à sa bonne place.
- 2- On recommence ces opérations pour la partie du tableau de 2 à n. Après quoi T[2] sera à sa bonne place, et ainsi de suite jusqu'à traitement de la partie formée par les deux derniers éléments.
Sachez que si les n-1 éléments du tableau sont bien placés le dernier le sera d'office.

```
Procédure tri_selection (VAR T : TAB, n : entier)
Variable i, j, x : entier
Début
    pour i de 1 à n - 1 faire
        pmin ← i
        pour j de i + 1 à n faire
            si T[j] < T[pmin] alors pmin ← j fin si
            si min ≠ i alors
                échanger (T[i], T[pmin])
            fin si
        fin pour
    fin pour
fin procédure
```

B. Tri à bulles:

La méthode de tri à bulles du tableau T consiste à :

- 1- Commencer par $i = 1$ et comparer $T[1]$ et $T[2]$ s'ils ne sont pas dans le bon ordre on les permute puis on passe à la paire $T[2]$ et $T[3]$ puis la paire $T[3]$ et $T[4]$ jusqu'à la dernière paire $T[n-1]$ et $T[n]$. On aura poussé ainsi le plus grand élément de T vers la dernière position du tableau, donc $T[n]$ est maintenant à sa bonne place donc plus la peine d'y revenir.
- 2- On recommence les opérations de l'étape 1 en parcourant de 1 à $n-2$ puis de 1 à $n-3$ et ainsi de suite. On arrête quand la partie à trier est réduite à un seul élément ou si lors du dernier parcours aucune permutation n'a été faite ; ce que signifie que le tableau est devenu trié.

```
Procédure tri_bulle (VAR T : TAB, n : entier)
Variable      i: entier
              échange_effectué : booléen
Début
  Répéter
    échange_effectué ← faux
    pour i de 1 à n - 1 faire
      si  $T[i] > T[i + 1]$ , alors
        échanger ( $T[i]$ ,  $T[i + 1]$  )
        échange_effectué ← vrai
      fin si
    fin pour
     $n \leftarrow n - 1$ 
  Jusqu'à (échange_effectué = faux) OU ( $n = 1$ )
fin procédure
```

C. Tri par insertion :

La méthode de tri par insertion du tableau T consiste à :

Nous supposons que les éléments de 1 à $i-1$ sont bien placés (triés), nous allons placer le $i^{\text{ème}}$ élément.

- 1- On commence par $i = 2$ et on cherche la position d'insertion de l'élément courant, à savoir celui de la position i , parmi les $i-1$ premiers éléments. Si cette position est i , l'élément est donc à sa bonne place, sinon, supposons que cette position est j . On décale d'un pas vers l'avant tous les éléments de j à $i-1$ puis on insère l'élément d'indice i à la position j .
- 2- On recommence les opérations de l'étape 1 jusqu'à ce qu'on atteigne la fin du tableau.

```
Procédure tri_insertion (VAR T : TAB, n : entier)
Variable i, j, x : entier
Début
    pour i de 2 à n faire
        x ← T[i]
        j ← i
        tant que j > 1 et T[j - 1] > x
            T[j] ← T[j - 1]
            j ← j - 1
        fin tant que
        T[j] ← x
    fin pour
fin procédure
```

Question :

- Essayer l'appliquer ces méthodes de tri sur le tableau T suivant : $T = [1, 7, 3, 0, 5, 2]$.

Les méthodes de recherche :**A. Recherche séquentielle :**

Cette méthode simple consiste à parcourir le tableau à partir du premier élément, et à s'arrêter dès que l'on trouve l'élément cherché (on ne cherche pas toutes les occurrences d'un élément). Soient T un tableau de n éléments et k l'élément qu'on recherche.

Question :

Essayer d'écrire l'algorithme d'une fonction nommée « recherche_seq » qui prend comme paramètres un tableau T d'entiers de taille N et un entier k . La fonction renvoie une valeur booléenne indiquant si l'élément k existe dans le tableau T ou non.

B. Recherche dichotomique :

Cette méthode s'applique si le tableau est déjà trié et s'apparente alors à la technique « Diviser pour Régner ».

Soient T un tableau déjà trié de n nombres et k le nombre qu'on recherche. On compare le nombre k au nombre qui se trouve au milieu du tableau T . Si c'est le même, on a trouvé, sinon on recommence sur la première moitié (ou la seconde) selon que k est plus petit (ou plus grand) que le nombre du milieu du tableau.

Question :

Essayer d'écrire l'algorithme d'une fonction nommée « recherche_dichotomique » qui applique la stratégie « Diviser pour Régner », bien qu'elle ne soit pas récursive, prend comme paramètres un tableau T d'entiers de taille N et un entier k . La fonction renvoie une valeur booléenne indiquant si l'élément k existe dans T ou non.