

Travaux dirigés

Langage C

Partie I : les chaînes de caractères

1. Ecrire un programme qui supprime la première occurrence d'une chaîne de caractère OBJ dans une chaîne SUJ.
2. Ecrire un programme qui remplace la première occurrence d'une chaîne CH1 par la chaîne CH2 dans une chaîne de caractère SUJ. Utiliser une chaîne de sauvegarder Fin pendant le remplacement.
3. Ecrire un programme qui remplace toutes les occurrences d'une chaîne de caractères CH1 par la chaîne CH2 dans une chaîne de caractères SUJ. Utiliser une chaîne de sauvegarde FIN pendant le remplacement.

Partie II : Le tri des tableaux

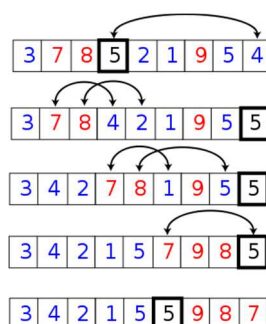
Le tri rapide (quick sort)

La méthode consiste à placer un élément du tableau (appelé pivot) à sa place définitive, en permutant tous les éléments de telle sorte que tous ceux qui sont inférieurs au pivot soient à sa gauche et que tous ceux qui sont supérieurs au pivot soient à sa droite.

Cette opération s'appelle le partitionnement. Pour chacun des sous-tableaux, on définit un nouveau pivot et on répète l'opération de partitionnement. Ce processus est répété récursivement, jusqu'à ce que l'ensemble des éléments soit trié.

Concrètement, pour partitionner un sous-tableau :

- on place le pivot à la fin (arbitrairement), en l'échangeant avec le dernier élément du sous-tableau ;
- on place tous les éléments inférieurs au pivot en début du sous-tableau ;
- on place le pivot à la fin des éléments déplacés.



Algorithme :

```
partitionner(tableau T, entier premier, entier dernier, pivot)
    échanger T[pivot] et T[dernier] // échange le pivot avec le dernier du
    tableau , le pivot devient le dernier du tableau
    j := premier
    pour i de premier à dernier - 1
        si T[i] <= T[dernier] alors
            échanger T[i] et T[j]
            j := j + 1
    échanger T[dernier] et T[j]
    renvoyer j

tri_rapide(tableau T, entier premier, entier dernier)
    début
        si premier < dernier alors
            pivot := choix_pivot(T, premier, dernier)
            pivot := partitionner(T, premier, dernier, pivot)
            tri_rapide(T, premier, pivot-1)
            tri_rapide(T, pivot+1, dernier)
        fin si
    fin
```