

# Algorithmique et programmation procédurale

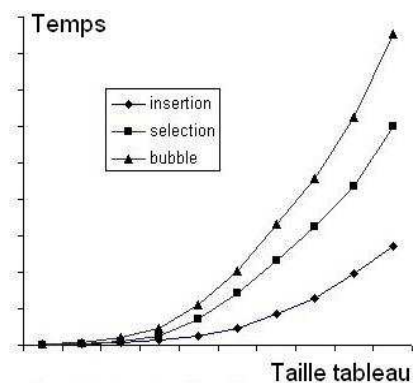
## TP4 – les Algorithmes de Tri

### Exercice 1

Nous connaissons les tris lents « sélection », « bulle » et « insertion » de complexité  $O(n^2)$ , ainsi que les tris rapides « quicksort » et « tri fusion » de complexité  $O(n \cdot \log n)$ . Implémentez ces algorithmes en C. Vous réalisez si possible toutes les cinq méthodes de tri, mais *au moins deux tris lents et quicksort*.

### Exercice 2

Nous allons effectuer une expérience qui permet de quantifier l'écart de performance entre les différents algorithmes de tri. L'idée est de chronométrer le temps utilisé pour trier des tableaux aléatoires de différentes tailles (par ex. 1000, 5000, 10000, 50000 ... cases) et d'afficher les résultats dans un graphe tel que



Cet exercice soulève trois questions techniques :

#### 1) Comment créer un tableau aléatoire ?

Etudiez le code suivant.

```
#include <stdlib.h>
#include <time.h>
int main ()
{
    srand (time(NULL)); // à quoi sert "srand"?
    printf ("Random number: %d\n", rand() % 10000); // à quoi sert "% 10000"?
    printf ("Random number: %d\n", rand() % 10000);
    printf ("Random number: %d\n", rand() % 10000);
    return 0;
}
```

## 2) Comment chronométrer les temps d'un algorithme ?

Etudiez le code suivant.

```
#include <time.h>
int main()
{
    clock_t debut, fin;
    int i;

    debut=clock();
    for (i=0;i<100000000;i++); // ici votre code à chronométrer
    fin=clock();

    float difference = (float)(fin-debut)/(float)CLOCKS_PER_SEC;
    printf("Durée : %f secondes\n", difference);
    return 0;
}
```

## 3) Comment afficher les statistiques enregistrées ?

Le plus simple sera d'enregistrer les temps chronométrés dans un fichier texte et de visualiser les graphes à l'aide d'un tableur (par exemple Excel ou OpenOffice). Pour écrire dans un fichier texte, voici un code qui vous inspirera.

```
#include <stdio.h>
int main()
{
    // ouvrir d'un fichier en écriture
    FILE *fic = fopen("test.txt", "w");
    if (fic == NULL) {
        printf("Impossible d'ouvrir le fichier!\n");
        return 1;
    }

    fputs("j'ecris\n", fic);
    fputs("dans un fichier!\n", fic);

    // fermeture du fichier
    if (fclose(fic) != EOF) {
        printf("Problème de fermeture du fichier!\n");
        return 1;
    }

    return 0;
}
```