

# TP - Le tri à bulles

## Objectif(s)

- ★ Travailler votre aisance avec la manipulation de valeurs dans un tableau
- ★ Programmer vos premiers algorithmes de tri et faire des mesures de performance.

## Exercices obligatoires

### Exercice 1 – Préliminaires

Dans cet exercice, vous allez écrire quelques fonctions sur les tableaux qui vous seront utiles par la suite. Tous les tableaux auront la même taille, définie dans une constante `TAILLE_MAX_TABLEAU` que vous serez amené à modifier plus tard. Commencez avec une taille de tableau petite. Par exemple 8.

#### Question 1

Écrivez ou récupérez une fonction d'affichage de tableau écrite lors d'un TP précédent. M Assurez vous qu'elle prend en compte des tableaux de taille `TAILLE_MAX_TABLEAU`.

#### Question 2

Écrivez une fonction `recopie_tableau` qui prend en paramètre deux tableaux de même dimension et qui recopie le premier tableau dans le deuxième.

Reprenez du TP précédent la fonction d'affichage d'un tableau et écrivez une fonction `main` qui contiendra les tests nécessaires pour valider votre fonction `recopie_tableau`.

#### Question 3

Écrivez une fonction `remplir_aleatoirement` qui prend en paramètre un tableau et qui remplit ses cases avec des nombres tirés au hasard.

Testez votre fonction `remplir_aleatoirement` en affichant le tableau une fois la fonction appelée.

### Exercice 2 – Evaluation des différentes versions de l'algorithme du tri à bulles

L'objectif de cet exercice est de programmer et de mesurer les performances des fonctions de tri à bulles qui ont été étudiées en TD.

#### Question 1

Programmez la première version du tri à bulles vue en TD (avec deux fonctions : `passage_bulle` et `tri_a_bullesV1`).

Vérifiez en affichant le tableau trié que votre fonction fonctionne correctement.

Modifier la taille du tableau en ajoutant une case et vérifiez que votre fonction de tri fonctionne toujours correctement.

#### Question 2

L'objectif de cette question est de mesurer le temps d'exécution de la fonction de tri. Le morceau de code suivant vous permet d'obtenir un temps d'exécution en seconde :

```
#include <time.h>

int main() {
    clock_t temps_initial, /* Temps initial en micro-secondes */
    temps_final;           /* Temps final en micro-secondes */
    double temps_exec;     /* Temps total en secondes */
    temps_initial = clock ();
```

```
... votre appel de fonction à insérer ici...
temps_final = clock ();
temps_exec = (temps_final - temps_initial) * 1e-6;
printf("temps d'execution %e\n", temps_exec);
```

Ajoutez la mesure du temps dans le programme de la question précédente. Vous mettrez aussi la ligne d’affichage des tableaux en commentaires.

Mesurez le temps d’exécution pour des tableaux de dimension 100, 1000, 10 000, 100 000, 1 000 000, 10 000 000, ... Arrêtez vous dès que le temps d’exécution dépasse les 30 secondes.

### Question 3

Choisissez la dernière taille  $n$  de tableau pour laquelle le temps d’exécution ne dépasse pas 10 secondes. Lancez votre programme sur les tailles correspondant à  $\{\frac{1}{10}n, \frac{2}{10}n, \frac{3}{10}n, \dots, \frac{9}{10}n, n\}$  Notez<sup>1</sup> les temps d’exécution et faites un graphique XY avec votre tableur favori représentant le temps d’exécution en fonction de la dimension du tableau. Le tri est-il un algorithme linéaire (c’est à dire dont le temps augmente proportionnellement avec la dimension du tableau)?

### Question 4

Programmez la deuxième version du tri à bulles vue en TD, et faites des mesures de performance et un graphique

### Question 5

Programmez la troisième version du tri à bulles vue en TD, et faites des mesures de performance et un graphique

## Renforcement

### Exercice 3

#### Question 1

Faites le meme travail que précédemment avec le tri sélection vu en TD.

Quel algorithme est le plus performant le tri à bulles ou le tri sélection ?

#### Question 2

Dans cette question, on souhaite observer le comportement de la 3e version du tri à bulles et du tri sélection sur des cas extrêmes.

Écrivez une fonction `remplir_ordre_croissant` qui remplit un tableau avec des valeurs de 1 à  $n$ ,  $n$  étant la dimension du tableau.

Écrivez une fonction `remplir_ordre_decroissant` qui remplit un tableau avec des valeurs de  $n$  à 1,  $n$  étant la dimension du tableau.

Mesurez les performances du tri à bulles et du tri sélection sur ces cas extrêmes.

#### Question 3

Quel algorithme choisir si vous avez à trier un tableau dont les valeurs sont :

- presque triées par ordre croissant
- presque triées par ordre décroissant

---

1. sur linux, vous pouvez copier une valeur en la sélectionnant, et la coller avec un click du milieu de votre souris.