# **Saé S1.02 :** Comparaison d'approches algorithmiques

## Programmes de tri d'un tableau

### Contexte

On dispose de données nationales sur le nombre de cas de Covid par tranche d'âge et par département de Bretagne. En vue de faciliter des recherche ou de fournir des résultats synthétiques, il peut être très utile de trier cet ensemble de données sur différents critères. Pour cela vous implémenterez plusieurs solutions et vous les comparerez.

### Présentation générale du projet

L'objectif de cette SAE est de comparer la performance de différents algorithmes de tri de tableaux.

Vous programmerez plusieurs algorithmes de tris appliqués à des tableaux d'entiers puis à des tableaux de chaînes de caractères, puis vous comparerez leurs performances selon certains critères.

Cette SAE est découpée en 5 étapes qui sont décrites à partir de la page suivante. À chaque étape, vous devrez déposer le résultat de votre travail sur Moodle selon le planning fourni en annexe. La SAE se terminera le 17 janvier, par un contrôle TP individuel.

### Organisation

Cette SAE est à faire en binôme.

Chaque binôme doit choisir un algorithme de tri dans chacune de ces 3 catégories :

| catégorie | algorithme |
|---|---|
| **facile** | *tri par sélection*<br>*tri par insertion*<br>*tri à bulles* |
| **un peu moins facile** | *tri shaker*<br>*tri à peigne* |
| **moins facile** | *tri rapide*<br>*tri fusion* |

ATTENTION : tout retard dans la remise des travaux entraînera 1 point en moins par jour de retard.

**Étape 1 : mise au point de vos programmes**

- Pour chaque algorithme de tri qui vous sont attribués, écrivez le programme correspondant en C. Votre programme doit être capable de trier un tableau d'entiers par ordre croissant des valeurs.

- Écrivez une version "décroissante" de chacun de vos programmes qui trie le tableau par ordre décroissant.

- Testez vos programmes sur un nombre restreint de valeurs (tableau constant de 10 entiers).
  Ne vous contentez pas d'un seul test d'exécution ! Vous devrez tester vos programmes de tri par ordre croissant dans le **cas moyen** (cas général) mais aussi dans le **cas favorable** (quand le tableau est déjà trié) et dans le **cas défavorable** (quand le tableau est trié par ordre décroissant.)

Rappel : un tableau peut être initialisé dès sa déclaration comme ceci :

```
#define MAX 10

int t[MAX] = {99, 47, 54, 12, 25, 39, 18, 26, 81, 39} ;
```

Pour chacun de vos tests vous fournirez une copie écran de l'exécution, conformément aux exemples suivants.

```
                TRI PAR SELECTION

tableau avant le tri :
 99  47  54  12  25  39  18  26  81  39

tableau apres le tri :
 12  18  25  26  39  39  47  54  81  99
```

```
                TRI PAR SELECTION

tableau avant le tri :
 12  21  21  23  34  46  59  67  67  98

tableau apres le tri :
 12  21  21  23  34  46  59  67  67  98
```

```
                TRI PAR SELECTION

tableau avant le tri :
 98  67  67  59  46  34  23  21  21  12

tableau apres le tri :
 12  21  21  23  34  46  59  67  67  98
```

**Étape 2 : changement d'échelle**

Écrivez de nouvelles versions de vos programmes : ils doivent désormais être capables de trier un tableau de 500 000 entiers. Pour cela vous écrirez une procédure qui remplit le tableau d'entiers avec des valeurs aléatoires comprises entre **1** et **10*RAND_MAX**.

Refaites les mêmes tests que dans la partie 1 (cas moyen, favorable et défavorable).

**Étape 3 : trier des chaînes de caractères**

Écrivez une troisième version de vos programmes pour qu'ils soient capables de trier un tableaux de 150 000 chaînes de caractères.

Vous écrirez une procédure qui remplit le tableau avec des chaînes construites de manière aléatoire et dont la longueur sera comprise aléatoirement entre 5 et 10 caractères. Vos chaînes ne seront composées que de lettres minuscules, donc comprises entre le caractère '`a`' (de code 97) et le caractère '`z`' (de code 122).

Refaites les mêmes tests que dans la partie 1 (cas moyen, favorable et défavorable).

**Étape 4 : mesures de performance – tableau d'entiers**

Pour pouvoir comparer l'efficacité de vos différents programmes de tri, vous allez mettre en place des indicateurs de performance :

- temps d'exécution du tri,
- nombre de comparaisons effectuées,
- nombre de permutations réalisées.

**Question 1**

Adaptez vos programmes de la partie 2 (tri de 500 000 entiers) afin qu'ils affichent en fin d'exécution la valeur de ces indicateurs (voir exemple ci dessous).

```
        TRI PAR SELECTION

Temps CPU du tri : 4.044 sec

Nombre de comparaisons : 1249975000

Nombre de permutations : 31775
```

Vous produirez une trace de chaque exécution et remplirez un tableau comme celui-ci :

Tri de 500 000 entiers : **cas Moyen**

| Algo | Temps exéc. | Nb comparaisons | Nb permutations |
|---|---|---|---|
| *Algo 1* | | | |
| *Algo 2* | | | |
| *Algo 3* | | | |

**IMPORTANT**

- Vous pourrez faire plusieurs essais et reporter la moyenne des résultats affichés
- Vous produirez de la même manière un tableau comparatif pour le cas **favorable** et un autre pour le cas **défavorable**.

**Question 2**

Adaptez vos programmes de la partie 3 (tri de 150 000 chaines de caractères) afin qu'ils affichent en fin d'exécution la valeur des trois indicateurs.
De la même façon que précédemment, établissez trois tableaux comparatifs.

**Étape 5 : données Covid**

Vous disposez sur Moodle d'un fichier contenant des "données Covid", c'est-à-dire des données sur plusieurs mois concernant le nombre de cas Covid recensés par jour et par tranches d'âge, dans les quatre départements bretons.

Chaque enregistrement (chaque "ligne" du fichier) possède la structure suivante :

```
struct int dep;         // n° du département (22, 29, 35 ou 56)

chaine date;      // date sous la forme aaaa-mm-jj
                  // exemple : 2020-05-23

int pos;          // nombre de cas enregistrés ce jour-là
                  // dans cette classe d'âge et dans ce
                  // département

int classe;       // tranche d'âge. La valeur 9 représente la
                  // tranche d'âge [0 - 9 ans], la valeur 19
                  // représente la tranche d'âge [10 – 19 ans],
                  // etc. jusqu'à la valeur 90 qui représente les
                  // 90 ans ou plus.
                  // Attention, la tranche d'âge 0 est un
                  // récapitulatif qui représente toutes les
                  // classes d'âge

int pop;          // nombre d'habitants de cette tranche d'âge
                  // dans ce département
```

À titre d'exemple, l'enregistrement suivant montre que le 20 octobre 2020, il y avait en Ille-et-Vilaine 81 cas de Covid chez les personnes agées entre 40 et 49 ans. Il montre aussi qu'il y 143 253 personnes de cette tranche d'âge qui vivent dans ce département.

```
35 2020-10-20 81 49 143253
```

Cet autre enregistrement, avec une classe égale à 0, indique que le 1er mai 2021, il y avait 82 cas de Covid en Ille-et-Vilaine, département qui compte 1 082 073 habitants.

```
35 2021-05-01 82 0 1082073
```

L'idée principale de cette étape est de copier les "données Covid" du fichier dans un tableau afin de pouvoir les trier sur différents critères et avant de programmer des recherches. Vous déclarerez et utiliserez un tableau de structures Covid.

**Question 1**
Dans un nouveau programme, écrivez une procédure `LireFichier` qui remplit un tableau de structures Covid passé en paramètre à partir du fichier *DonneesCovid.data,* et qui fournit en paramètre de sortie le nombre d'enregistrements effectivement lus.

Remarque : il y a 17 776 "lignes" dans le fichier, c'est-à-dire 17 776 enregistrements Covid.

**Question 2**
Écrivez une procédure `triDepartement(...)` qui trie le tableau des données Covid sur le **département**. Vous utiliserez le programme de tri de votre choix.

**Question 3**
Écrivez une fonction `population22(...)` qui retourne le nombre d'habitants des Côtes d'Armor.

**Question 4**
Écrivez une procédure `triDate(...)` qui trie le tableau des données Covid passé en paramètre, sur la **date**. Vous utiliserez le programme de tri de votre choix.

**Question 5**
Écrivez une procédure `DebutAnnee35(…)` qui affiche les données par tranche d'âge en Ille-et-Vilaine lors du 1er janvier 2021, conformément à cet exemple :

```
Nombre de cas en Ille-et-Vilaine par classe d'age, le 1er janvier 2021

[ 0 -  9] : 0
[10 - 19] : 0
[20 - 29] : 1
[30 - 39] : 3
[40 - 49] : 1
[50 - 59] : 2
[60 - 69] : 0
[70 - 79] : 1
[80 - 89] : 1
[90 - 99] : 3

 TOTAL : 12
```

Vous conserverez les résultats dans un tableau intermédiaire avant de les afficher.

**ANNEXE : planning de vos remises de travaux**

Afin de juger de l'avancée de votre travail vous devrez, au cours de votre projet, fournir les éléments indiqués dans ce planning.

**Planning**

| Quand | Quoi | Où |
|---|---|---|
| **8 décembre** | Code source de vos programmes de la partie 1<br><br>Traces d'exécution : affichage des valeurs du tableau avant le tri et affichage après le tri | sur Moodle |
| **15 décembre** | Code source de vos programmes de la partie 2 | sur Moodle |
| **22 décembre** | Code source de vos programmes de la partie 3 | sur Moodle |
| **9 janvier** | Code source de vos programmes de la partie 4<br><br>Tous vos tableaux comparatifs | sur Moodle |
| **16 janvier** | Code source de votre programme de la partie 5 | sur Moodle |
| **17 janvier** | Contrôle TP individuel | |