

TP 10 - Tri d'éléments

Dans ce TP, nous allons comparer différentes méthodes de tri. Pour cela, nous allons considérer le problème de tri des données situées dans la première colonne de la feuille de calcul Excel.

Q1 Écrire la macro **generation** qui remplit les cellules de la colonne L1C1:L500C1 par 500 nombres compris entre 1 et 10000 générés aléatoirement. On pourra utiliser pour cela la fonction **Rnd** de VBA qui génère un nombre aléatoire compris entre 0 et 1. Pour générer un entier compris entre a et b il suffit donc d'exécuter l'instruction :

```
Int((b - a + 1)* Rnd() + a)
```

Exécuter la macro **generation** pour remplir la colonne 1.

Q2 Écrire la macro **TriSelect** qui place les éléments de la première colonne de la feuille de calcul dans la seconde colonne dans l'ordre croissant. Pour cela, **TriSelect** copie les éléments de L1C1:L500C1 dans L1C2:L500C2, puis applique l'algorithme de tri par sélection pour trier les éléments de la deuxième colonne.

Q3 Écrire une fonction **verifTri** qui prend en paramètre un numéro de colonne j et retourne vrai si les éléments de la colonne j , de la ligne 1 à la ligne 500, sont triés en ordre croissant, et faux sinon. Puis, écrire une macro qui vérifie que le tri réalisé par **TriSelect** est correct en utilisant **verifTri**.

Q4 Écrire la macro **TriBulle** qui place les éléments de la première colonne de la feuille de calcul dans la troisième colonne dans l'ordre croissant. Pour cela, **TriBulle** copie les éléments de L1C1:L500C1 dans L1C3:L500C3, puis applique l'algorithme de tri à bulles pour trier les éléments de la troisième colonne.

Q5 Écrire la macro **verifieTris** qui affiche un message d'erreur si le résultat de l'exécution de **TriSelect** et **TriBulle** n'est pas le même ou si les éléments ne sont pas triés en ordre croissant. Tester la macro **verifieTris**, et corriger si nécessaire.

Q6 On s'intéresse maintenant à la comparaison du temps d'exécution des ces deux méthodes de tri. Pour cela écrire une macro **Test** qui exécute successivement chacune des deux méthodes en mesurant leur temps d'exécution. La macro **Test** affiche ensuite les deux temps d'exécution obtenus. Pour mesurer le temps d'exécution des méthodes, on pourra utiliser la fonction **Timer** de VBA qui retourne le temps système courant. Ainsi pour mesurer le temps d'exécution d'une méthode **execMethode**, il suffit d'écrire les instructions suivantes :

```
Dim debut As Single , fin As Single  
debut = Timer  
execMethode  
fin = Timer
```

Le calcul **fin - debut** permet de connaître le temps d'exécution de **execMethode**. Quelle est la méthode de tri la plus efficace ? Pourriez-vous expliquer pourquoi ? Tester aussi avec des données initialement triées ou quasiment triées, et des données initialement triées en ordre inverse.