

## TP6

**Fichiers à récupérer** Depuis l'*Espace pédagogique* récupérez les fichiers `mainTP6.cpp`, `outilsTab6.cpp`, `outilsTab6.h`, ainsi que les fichiers `trace1.gnu` et `trace2.gnu`.

Le fichier `mainTP6.cpp` contient le `main`.

Le fichier `outilsTab6.cpp` contient les définitions de fonctions sur les tableaux d'entiers (affichage, génération aléatoire, chronométrage de fonctions) ainsi que des fonctions que vous devrez compléter. Leurs spécifications sont données dans le fichier entête `outilsTab6.h`.

Pour compiler : `g++ -Wall -ansi -pedantic mainTP6.cpp outilsTab6.cpp -o tp6`

Les fichiers `traceX.gnu` contiennent des directives `gnuplot` pour l'affichage de courbes.

## Algorithmes de Tri

Il s'agit de comparer les différents algorithmes de tri par comparaison étudiés en cours/TD : tri par insertion, tri par sélection, tri par tas, tri rapide, tri par fusion.

### Programmer les algorithmes de tri

Complétez dans le fichier `outilsTab6.cpp` les fonctions de tri :

- tri par insertion, vu en cours
- tri par sélection. Pour ceux qui ne l'auraient pas encore étudié, son principe est simple : on calcule le plus petit élément de `T[0..taille-1]` que l'on échange avec `T[0]`, puis on calcule le plus petit élément de `T[1..taille-1]` que l'on échange avec `T[1]`, ainsi de suite.
- tri par tas, vu en cours : la fonction est déjà définie. Vous n'avez rien à faire.
- tri rapide étudié en cours
- tri par fusion étudié en TD.

Vérifiez que toutes ces fonctions donnent le même résultat en exécutant le programme avec l'option 1. Celui-ci applique les différentes fonctions de tri sur un tableau aléatoire de 10 entiers et affiche leur résultat.

### Comparaison des temps d'exécution des algorithmes de tri

Exécutez le programme avec l'option 2. Les 5 fonctions sont exécutées pour des tableaux aléatoires dont la taille varie de 2000 à 20000 et les courbes de leur temps d'exécution sont tracées. Vous devez constater que les courbes des 2 algorithmes de tri de complexité quadratique (`triInsertion` et `triSélection`) se détachent nettement des 3 autres.

Dans un second temps les 3 fonctions de tri les plus rapides (`triFusion`, `triParTas` et `triRapide`) sont exécutées pour trier des tableaux aléatoires de taille comprise entre 50000 et 500000. Comparez les 3 courbes.

### Nombre de valeurs différentes

Complétez dans le fichier `outilsTab6.cpp` la fonction `nbValDiff(Tab,t)` qui doit renvoyer comme résultat le nombre de valeurs différentes des éléments du tableau `Tab` de taille `t`. Par exemple le tableau 

2	7	7	9	2	7
---	---	---	---	---	---

 contient 3 valeurs différentes.

Quelle est la complexité de votre fonction ?

Exécutez le programme avec l'option 3. Celui-ci applique la fonction `nbValDiff` pour un tableau aléatoire d'un million d'entiers. Si votre algorithme est efficace, le temps de réponse ne doit pas excéder quelques secondes.