

TP 1 : Comparaison de temps de calcul d'algorithmes de tri

Objectif

Pour tout ce TP, on se créera trois tableaux, l'un de longueur 10 (pour les tests de validité), les deux autres de longueur 1000 et 10000 (pour les tests d'efficacité), remplis de nombres aléatoires compris entre 0 et 10000, et on déterminera le temps d'exécution des algorithmes de tri sur ces 2 tableaux, de sorte à comparer les performances des différents algorithmes.

1. Tri par sélection

Le tri par sélection consiste à rechercher le plus petit élément du tableau initial, et à le mettre en position initial (par exemple par un échange), puis à recommencer, avec la recherche du plus petit terme parmi ceux restants etc.

2. Tri par fusion (Merge Sort)

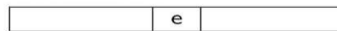
À partir de deux listes triées, on peut facilement construire une liste triée comportant les éléments issus de ces deux listes (fusion).

Le principe de l'algorithme de tri fusion repose sur cette observation : le plus petit élément de la liste à construire est soit le plus petit élément de la première liste, soit le plus petit élément de la deuxième liste. Ainsi, on peut construire la liste élément par élément en retirant tantôt le premier élément de la première liste, tantôt le premier élément de la deuxième liste (en fait, le plus petit des deux, à supposer qu'aucune des deux listes ne soit vide, sinon la réponse est immédiate).

3. Tri rapide (Quicksort)

- Est de découper (rapidement) le tableau en deux parties, une partie (T-) où tous les éléments sont inférieurs à une valeur donnée (le pivot) e et l'autre partie (T+) où tous les éléments sont supérieurs ou égaux au pivot.
- Ensuite on appelle récursivement Quicksort indépendamment sur chaque partie.
- Il est facile de voir qu'un tel processus garantit que le tableau résultant est trié, du moment que les parties diminuent en taille.
- Cela se fait en définissant trois fonctions :
 - ❖ **La fonction TrouvePivot** qui recherche un pivot valide,
 - ❖ **La fonction Partition** qui découpe le tableau, et
 - ❖ **La fonction Quicksort** qui appelle les deux autres et s'appelle récursivement.

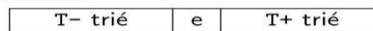
- Choisir arbitrairement un élément e :



- Constituer les sous listes T- et T+ (en temps linéaire) :



- Recommencer récursivement avec les tableaux T- et T+ puis reconstituer T :



- Le tableau T est alors trié.

Travail à faire :

Développez un programme permettant le tri, et donner une comparaison de la complexité en terme de temps de calcul (rapidité) et d'espace mémoire nécessaire pour l'exécution.