TP 2 Algorithmique Avancée : Tri par Tas

CERI - Licence 2 Informatique

Semestre 3

Le TP consiste à implémenter et tester dne classe, nommée "tableau". Elle permet de générer aléatoirement des tableaux d'entiers et de comparer les performances de 3 tris.

1/ Créer un fichier tp2.h contenant les déclarations des classes suivantes.

- class tableau
  {
  int* T ;
  int n ;
  public :
  tableau(int n) ;
  ∼tableau() ;
  void affiche() ;
  int triTas() ;
  int triBulle() ;
  int triInsertion() ;
  int test(int n) ;
  }

— T : tableau d'entiers à trier créé grâce au constructeur.
— n : nombre d'éléments de T
— tableau(int n) : constructeur créant un tableau de taille n.
— affiche() : affichage du contenu de T.
— triBulle() : trie le tableau T avec un tri bulle et renvoie le nombre d'échanges d'entiers effectué.

— triInsertion() : trie le tableau T avec un tri par insertion et renvoie le nombre d'échanges d'entiers effectué.
— triTas() : trie le tableau T avec un tri par tas et renvoie le nombre d'échanges d'entiers effectué.

Le tri pas tas nécessite des méthodes additionnelles. Il faudra rajouter (au moins) les deux suivantes :

— void reorganiser(int j) : permettant de ré-organiser T à partir de l'indice j si ce n'est pas un tas.
— int suppression() : renvoie l'élément minimum et réorganise T pour qu'il reste un tas. Cette méthode suppose que T est un tas, ou a été réorganisé comme tel.

La méthode "int test(int n)" :

— effectue 10 itérations.
— Dans chacune d'elle, n entiers dans $[100, 2000]$ sont générés aléatoirement et affectés à T,
— Puis, les 3 tris sont lancés sur le tableau obtenu. Il faudra prévoir une copie de T, dans un tableau auxiliaire, pour que les 3 tris aient lieu sur le même tableau.

Les résultats de ces 10 itérations doivent être affichés sous forme de tableau contenant les colonnes suivantes [1],

n	Nom du Tri	Nombre d'échanges	Temps d'exécution
...	...	...	...

Le temps d'exécution ne doit compter que le temps effectif du tri, pas celui de la copie dans un tableau auxiliaire.

Dans votre programme principal, pour chaque valeur $n = 10, 100, 1000, ..., 10^6$:
— vous créerez un objet de type "Tableau",
— puis vous lancerez la méthode "test()"

1. Pour mesurer le temps, utilisez le type C++ "$time\_t$", ainsi que les fonctions $time(.)$ et $difftime(.,.)$