Devoir I

8INF870 - Algorithmique

Aymen Sioud

UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

Département d'Informatique et de Mathématique

Consignes

- **A remettre au plus tard le 18 mai 2016 à 19h (heure de l'est).**
- **Le travail devra se faire individuellement.**
- **Le travail devra être remis par courriel à l'adresse du professeur en un seul répertoire compressé : Aymen_Sioud@uqac.ca.**
- **En plus d'un rapport contenant les différents algorithmes et les livrables de l'exercice de Tri, vous devrez remettre un seul projet qui contiendra les différents exercices**
- **Le langage utilisé est C++**
- **Chaque fichier devra comporter un entête avec votre nom et une brève description**
- **Vous êtes libre de choisir une exécution en ligne de commande ou non**
- **Le travail remis devra être nommé : Devoir1_nom1_8INF870_E2016.**

Multiplication Matricielle : Algorithme de Strassen

Écrire en C++ le programme qui implémente l'algorithme de Strassen pour le calcul du produit de deux matrices carrées d'ordre *n.*

- Concevoir une classe « Matrice » pour représenter et manipuler des matrices carrées d'ordre *n*.
- Implémenter l'algorithme cubique naïf qui calcule le produit de deux matrices carrées d'ordre *n*.
- Vous devrez surcharger l'opérateur × pour l'algortihme de Strassen.

Tri

Écrire en C++ un programme qui pourra comparer les algorithmes des tris suivants :

- Tri par fusion
- Tri par bulles
- QuickSort
- QuickSort rand (pivot choisi aléatoirement)
- Tri par insertion
- Tri par tas
- Tri par base

Vous devrez générez un tableau contenant *n* éléments aléatoirement générés entre 0 et INT_MAX (`limits`). Ce tableau sera utilisé pour tous les algorithmes.
Pour le temps d'exécution utiliser la fonction clock(). Chaque algorithme devra être exécuté 10 fois.
Vous devrez remplir le tableau ci-dessous.
Tracez les courbes correspondantes et essayez de trouver des fonctions de complexités rendant compte de vos résultats.

Tps moyen	*n*	*2n*	*3n*	*5n*	*7n*	*10n*	*100n*	*10n trié*	*100n trié*	*10n trié inversé*	*100n trié inversé*
Tri par fusion											
Tri par bulles											
QuickSort											
QuickSort rand											
Tri par insertion											
Tri par tas											
Tri par base											
sort											
sort_heap											
stable_sort											

N.B : sort, sort_heap et stable_sort sont les algorithmes de la STL.

Grand nombre : Algorithme de Karatsuba

Écrire en C++ le programme qui implémente l'algorithme de Karatsuba pour le calcul du produit de deux entiers ayant *n* chiffres.

- Concevoir une classe « grandEntier » pour représenter et manipuler les grands entiers.
- Surcharger les opérateurs « + » et « * » pour cette classe et utiliser l'algorithme de Karatsuba pour effectuer le calcul.