Usare almeno due funzioni

ASD 23/24 - Laboratorio 2

In questo laboratorio vi viene richiesto di implementare in maniera ricorsiva l'algoritmo QuickSort con partizionamento in place, considerando due diverse strategie per la scelta del pivot:

1. **Scelta banale:** il pivot è sempre il primo elemento della porzione di array che stiamo considerando;
2. **Scelta casuale:** la posizione del pivot viene scelta casualmente tra quelle disponibili nella porzione di array che stiamo considerando.

Nella seconda parte del laboratorio, vi viene richiesto di condurre un'analisi sui tempi di esecuzione di QuickSort e degli altri algoritmi di ordinamento visti a lezione (la cui implementazione vi viene fornita).

1 Materiale dato

Nel file `asd-lab2-traccia.zip`, trovate:

- Un file `ASD-sort.h` contenente le intestazioni di diverse funzioni di ordinamento in place di un `vector` di interi
- Un file `ASD-sort.cpp` dove dovete scrivere l'implementazione delle funzioni richieste
- Un file `ASD-main-sort.cpp` contenente un programma principale per testare le funzioni
- Diversi file `.txt` che contengono sequenze di numeri interi e possono essere utilizzati come file di input
- Un file `tempi-di-esecuzione-algo-ordinamento.xls` è un file excel che vi consente di tenere traccia dei tempi di esecuzione dei diversi algoritmi. In particolare, vi viene richiesto di riportare sul file, nelle colonne appropriate, i tempi di tre diverse esecuzioni. La media dei tempi delle tre esecuzioni, che dovete considerare nella successiva analisi, viene calcolata in automatico.

Gli unici file da modificare sono quindi `ASD-sort.cpp` e `tempi-di-esecuzione-algo-ordinamento.xls`.

2 Funzioni da implementare

Il file `ASD-sort.h` contiene i prototipi delle funzioni che andranno implementate da voi nel file `ASD-sort.cpp` e richiamate in `ASD-main-sort.cpp`. In `ASD-sort.cpp` troverete anche implementazione di altre funzioni di ordinamento che non vi chiediamo di implementare.

Vi chiediamo di implementare le funzioni seguenti (solo ed esclusivamente), contenute nel file `ASD-sort.cpp`). **NOTA:** Gli altri file non devono essere modificati (salvo che per scopi di testing, ma poi devono essere ripristinati come da originale).

```
/************************************************/
/*       prototipi di funzioni da implementare    */
/************************************************/

/*quicksort con scelta banale del pivot*/
void quickSortTrivial(vector<int>& v);

/*quicksort randomizzato*/
void quickSortRandom(vector<int>& v);
```

Ogni volta, che completate una funzione, vi raccomandiamo di compilare il file usando il comando:
`g++ -Wall -std=c++14 -c ASD-sort.cpp`
per verificare gli errori di sintassi. Se lo ritenete necessario, potete crearvi un vostro programma main per eseguire dei vostri test.

Vi ricordiamo che un programma per generare un numero random fra 0 e 100 escluso è il seguente:

```
#include <cstdlib>
#include <ctime>

int main() {
  srand(time(NULL)); // questa linea va eseguita una volta sola nel programma
  int randomNumber = (rand()%100); // questa invece va eseguita ogni volta
                                   //che vogliamo generare un nuovo numero
}
```

N.B.: La chiamata `srand(time(NULL));` è gia scritta nel programma principale!

3 Programma principale

Il file `ASD-main-sort.cpp` contiene il `main` di un programma per aiutarvi a svolgere dei tests. Questo programma richiede una lista di interi contenuta in un file e esegue i diversi algoritmi di ordinamento e per ciascun algoritmo stampa il tempo di esecuzione in microsecondi. In più il programma verifica che la vostra implementazione produca il risultato corretto.
Per potere usare questo programma, potete compilarlo come segue:

```
g++ -Wall -std=c++14 ASD-sort.cpp ASD-main-sort.cpp -o ASD-main-sort
```

e poi eseguirlo con `./ASD-main-sort`.

Dovete usare questo programma per ottenere i valori da inserire nel file `tempi-di-esecuzione-algo-ordinamento.xls`.

4 Consegna

Per la consegna, creare uno `zip` con tutti i file forniti; in particolare con il file `ASD-sort.cpp` da voi modificato e il file `tempi-di-esecuzione-algo-ordinamento.xls` completato.