

Laboratorio di Algoritmi e strutture dati Esercizio 1

Giorgio Mecca
Mat: 880847

```
giorgio@giorgio-VirtualBox:~/Scrivania/laboratorio-algoritmi-2019-20/Esercizio1$ make
cc -g -O0 -I include -I Resources/C/Unity/ -c src/sorting_G.c -o sorting_G.o
cc -g -O0 -I include -I Resources/C/Unity/ test/sorting_G_test.c sorting_G.o Resources/C/Unity/unity.c -o testoo
cc -g -O0 -I include -I Resources/C/Unity/ Sorting_G_app/sorting_G_app.c sorting_G.o -o sorting_G_app
giorgio@giorgio-VirtualBox:~/Scrivania/laboratorio-algoritmi-2019-20/Esercizio1$ ./sorting_G_app records.csv
Insertion sort con field1:
    Esecuzione algoritmo fallita

Insertion sort con field2:
    Esecuzione algoritmo fallita

Insertion sort con field3:
    Esecuzione algoritmo fallita

Quick sort con field1:
    Esecuzione algoritmo fallita

Quick sort con field2:
    Tempo impiegato :      31 secondi

Quick sort con field3:
    Tempo impiegato :      36 secondi

giorgio@giorgio-VirtualBox:~/Scrivania/laboratorio-algoritmi-2019-20/Esercizio1$
```

L'immagine riportata sopra rappresenta l'esecuzione dell'esercizio 1. Questo prevedeva la realizzazione di una libreria C contenente l'insertion Sort e il quickSort ("sorting_G"), in seguito è stato realizzato un applicativo che effettuava dei test con il file "records.csv" su questi due algoritmi. Il file conteneva dei record(Intero id, Stringa field1, Intero field2, Floating Point field3),circa 20'000'000, con cui si è effettuato l'ordinamento con entrambi gli algoritmi su ognuno dei 3 campi field (6 esecuzioni in totale) riportando per ogni esecuzione il tempo necessario, stampa "Esecuzione algoritmo fallita" se questa supera i 10 min. Possiamo notare che l'insertion sort fallisce in ogni caso, questo perché ha un tempo di esecuzione pari a $O(n^2)$ che lo rende molto veloce sugli ordinamenti di pochi elementi, estremamente lento su grandi valori come nel nostro caso(20 000 000). Il QuickSort invece ha tempo di esecuzione pari a $O(n \log n)$ e quindi riesce a ordinare anche grandi quantità di elementi in 31/36 secondi. Il quicksort però fallisce con il campo field1 (tipo Stringa) poiché possiamo ipotizzare che la funzione di confronto delle stringhe "strcmp()" peggiori notevolmente la complessità dell'algoritmo.