RELAZIONE LABORATORIO ALGORITMI 2017/2018

ESERCIZIO 1:

Nel primo esercizio abbiamo sviluppato una libreria chiamata **“sort”** in cui abbiamo sviluppato due algoritmi di ordinamento, nello specifico abbiamo implementato **Merge Sort** e **Insertion Sort**. Inoltre abbiamo anche sviluppato un metodo **Sum** che presi in input un intero N e un array di interi verifica se nell’array è contenuta la somma di almeno 2 elementi che sia uguale a N.

Sort:

La libreria è stata pensata per lavorare con tipi generici, quindi abbiamo usato la classe Comparator, e abbiamo creato 3 classi (**IntegerComparator, LongComparator, StringComparator**). Il costruttore della classe Sort quindi richiede in ingresso un oggetto di tipo Comparator così da sapere con quale criterio ordinare i valori.

Inoltre abbiamo sviluppato altri due metodi, **swap** e **printArray**. Il primo scambia due valori date le posizioni e l’altro stampa l’array.

Sort Runner:

Nel package sortrunner abbiamo sviluppato l’usage, per testare il funzionamento di InsertionSort, MergeSort e Sum con i dataset forniti.

Sum:

Package in cui abbiamo inserito la classe Sum.java contenente il metodo sum.

Test:

Abbiamo sviluppato i test in due cartelle separate.

Considerazioni sulla complessità degli algoritmi sviluppati:

Abbiamo iniziato a sviluppare inizialmente l'algoritmo **InsertionSort**, di complessità quadratica, e abbiamo provato ad ordinare il file Integers.txt contenente circa 20 milioni di numeri. Provando ad ordinare il file abbiamo notato che l'algoritmo, data la sua elevata complessità, non riuscirà ad ordinare il file in un tempo considerevolmente piccolo, infatti dopo 10 minuti di lavoro l'algoritmo non è riuscito ad ordinare l'array.

Ripetendo l'esperimento per il **MergeSort**, che ha complessità [n log(n)], abbiamo invece osservato che l'algoritmo riesce ad ordinare il file in circa 40 secondi.

Sviluppando l'algoritmo Sum invece abbiamo dovuto tenere conto della complessità come specificato dal testo del progetto, infatti avremmo potuto facilmente sviluppare un algoritmo quadratico con due cicli for annidati. Solo che l'algoritmo se testato con il file Integers.txt avrebbe avuto dei lunghissimi tempi d'attesa. Sviluppando l'algoritmo con una complessità [K log K] con K numero di elementi nell'array passato al metodo siamo riusciti a verificare in breve tempo la presenza di somme nel file.