Esercizio 1

Alessandro D'Amico

3 Luglio 2019

Indice

1 Introduzione al problema

Nel seguente esperimento viene preso in considerazione il problema dell'ordinamento, per la cui soluzione utilizziamo due differenti algoritmi: Insertion Sort e Merge Sort. Lo scopo e' ottenere un array i cui elementi che lo compongono (inizialmente disposti in modo casuale) siano disposti in ordine crescente.

2 Caratteristiche teoriche di algoritmi e e strutture utilizzate

Insertion

3 Prestazioni attese

4 Esperimenti

5 Documentazione del codice

6 Risultati

Sono stati effettuati i test con piu set di numeri (set da 10, 100, 1000 valori), tenendo conto del fatto che la dimensione di ogni singolo numero (in termini di cifre) non incide sulla nostra analisi, trattandosi di algoritmi per confronto (non si tiene conto del max o del min). E' dunque necessario distinguere due casi, a seconda che si usi un set ridotto o un set di grandi dimensioni.

6.1 Per pochi input

Dalla Figura 1 traspare che per ordinare meno di 80 numeri Insertion Sort risulta piu' veloce (cio' e' dovuto principalmente al tempo impiegato al caricamento delle funzioni sulla stack).

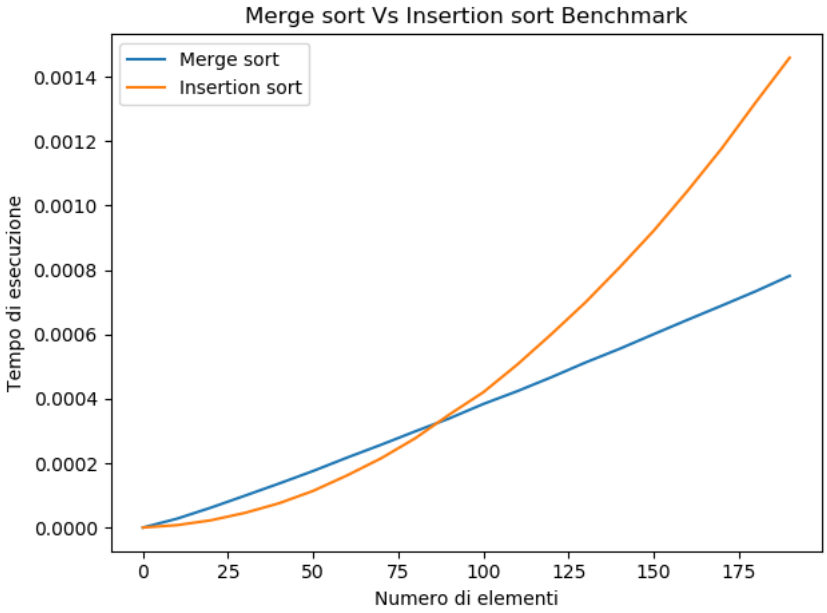

Figura 1: Merge Sort vs Insertion Sort per pochi input

6.2 Per tanti input

Grazie alla figura 2 e' facilmente deducibile che per grandi set di numeri da ordinare Merge Sort e' sensibilmente piu' rapido rispetto a Insertion Sort

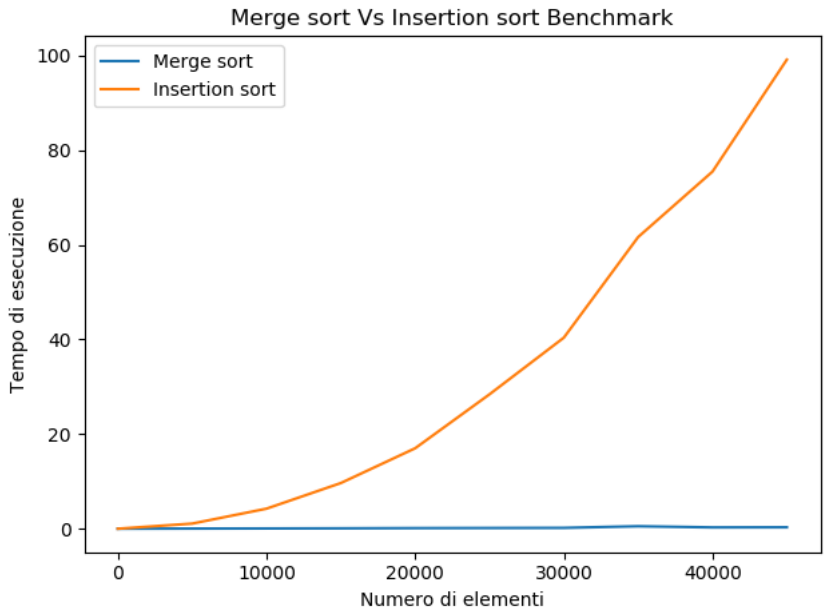

Figura 2: Merge Sort vs Insertion Sort per molti input

7 Conclusioni

E' stato verificato il comportamento asintotico di Insertion Sort e Merge $\longrightarrow \infty$