

Autori : Lobascio Giuseppe Pio, MATR: 916981, Mansi Matteo, MATR: 929683

Relazione Esercizio 1

Questo esercizio permette di effettuare gli algoritmi di BinaryInsertionSort e MergeSort su 3 campi differenti presi dal file CSV.

Abbiamo eseguito i test con diversi valori di k, riscontrando delle notevoli differenze tra il test effettuato usando il BinaryInsertionSort e con il MergeSort in termini di tempo di esecuzione.

Si riportano sotto alcuni dei test effettuati.

K = 10

Test effettuato con BinaryInsertionSort :

test on int array passed in 0.00001 seconds

test on float array passed in 0.00002 seconds

test on string array passed in 0.00002 seconds

Test effettuato con MergeSort :

test on int array passed in 0.02699 seconds

test on string array passed in 0.03102 seconds

test on float array passed in 0.02874 seconds

K = 100

Test effettuato con BinaryInsertionSort :

test on int array passed in 0.00012 seconds

test on string array passed in 0.00014 seconds

test on float array passed in 0.00011 seconds

Test effettuato con MergeSort :

test on int array passed in 0.02615 seconds

test on string array passed in 0.03166 seconds

test on float array passed in 0.02933 seconds

K = 10000

Test effettuato con BinaryInsertionSort :

test on int array passed in 0.07410 seconds

test on string array passed in 0.05961 seconds

test on float array passed in 0.08196 seconds

Test effettuato con MergeSort :

test on int array passed in 0.01282 seconds

test on string array passed in 0.01813 seconds

test on float array passed in 0.00722 seconds

K = 200000

Test effettuato con BinaryInsertionSort :

test on int array passed in 20.19505 seconds

test on string array passed in 20.07360 seconds

test on float array passed in 20.21720 seconds

Test effettuato con MergeSort :

test on int array passed in 18.32103 seconds

test on float array passed in 18.74536 seconds

test on string array passed in 16.39380 seconds

L'ordinamento sia effettuato tramite BinaryInsertionSort che tramite MergeSort si protrae più di 10 minuti nel caso in cui $k \geq \underline{1500000}$.

I risultati sono quelli che ci aspettavamo e abbiamo notato come fino ad un $k <$ della metà delle linee del file, i tempi sono molto vicini tra loro. La differenza tra i due algoritmi di ordinamento nei tempi stampati ci porta ad asserire che il BinaryInsertion è più rapido con un $k < 10000$, raggiunge la sua massima efficienza con un k tra il 25 e 100...diventando più lento specialmente da $k \geq 200000$, viceversa il MergeSort.