

# Università degli Studi di Torino

## Relazione Laboratorio di Algoritmi

### ESERCIZIO 1

Chironna Luca - Ferrero Regis Riccardo – Ferrod Roger

## INTRODUZIONE

Lo scopo di questo esercizio era quello di implementare i seguenti tipi di algoritmi di ordinamento, Insertion Sort, Selection Sort, Quick Sort, per poi utilizzarli su un file contenente 20 milioni di records da ordinare.

I tempi di risposta di questi algoritmi sono stati presi tramite la funzione clock() su una macchina del laboratorio avente le seguenti caratteristiche :

- Scientific Linux 6.8(Carbon)
- 7.5 GiB of Memory
- 8 Processors Intel Core i7-2600 CPU @3.40GHz

Per quanto visto a lezione ci aspettavamo che il Quick Sort, avente complessità temporale media di  $\Theta(n \log n)$  sarebbe stato il più veloce tra i tre, avendo sia Insertion Sort che Selection Sort una complessità di  $\Theta(n^2)$ .

Abbiamo implementato il Quick Sort con la possibilità di modificare la posizione del pivot nella partizione per confrontare i tempi di ordinamento di tutti i casi possibili.

Gli elementi del record sono caricati in memoria senza un pre-ordinamento, ogni riga è quindi in una posizione randomica.

I tempi sono stati calcolati sul primo campo, in quanto due stringhe di caratteri sono più lente da ordinare rispetto a due numeri a causa del confronto carattere per carattere.

## RACCOLTA DATI

<i>Insertion Sort</i>	N	Load(s)	t(s)
	1000	0	0,00
	10000	0	0,265
	50000	0,2	7,615
	100000	0,04	34,97
	150000	0,055	116,635
	200000	0,085	306,605
	250000	0,1	513,695

<i>Selection Sort</i>	N	Load(s)	t(s)
	1000	0	0,00
	10000	0	0,62
	50000	0,02	16,365
	100000	0,04	73,145
	150000	0,06	241,75
	200000	0,09	508,42
	250000	0,105	853,69

***Quick Sort  
Random Pivot***

<b>N</b>	<b>Load(s)</b>	<b>t(s)</b>
<b>1000</b>	<b>0</b>	<b>0,00</b>
<b>10000</b>	<b>0</b>	<b>0,01</b>
<b>50000</b>	<b>0,015</b>	<b>0,02</b>
<b>100000</b>	<b>0,04</b>	<b>0,03</b>
<b>150000</b>	<b>0,06</b>	<b>0,05</b>
<b>200000</b>	<b>0,08</b>	<b>0,065</b>
<b>250000</b>	<b>0,11</b>	<b>0,075</b>
<b>500000</b>	<b>0,21</b>	<b>0,175</b>
<b>1000000</b>	<b>0,42</b>	<b>0,42</b>
<b>5000000</b>	<b>2,285</b>	<b>2,775</b>
<b>10000000</b>	<b>4,41</b>	<b>6,365</b>
<b>15000000</b>	<b>6,435</b>	<b>10,18</b>
<b>20000000</b>	<b>8,35</b>	<b>14,41</b>

***Quick Sort  
Median Pivot***

<b>N</b>	<b>Load(s)</b>	<b>t(s)</b>
<b>1000</b>	<b>0</b>	<b>0,00</b>
<b>10000</b>	<b>0</b>	<b>0,01</b>
<b>50000</b>	<b>0,025</b>	<b>0,015</b>
<b>100000</b>	<b>0,045</b>	<b>0,025</b>
<b>150000</b>	<b>0,06</b>	<b>0,05</b>
<b>200000</b>	<b>0,085</b>	<b>0,06</b>
<b>250000</b>	<b>0,11</b>	<b>0,075</b>
<b>500000</b>	<b>0,21</b>	<b>0,165</b>

<b>1000000</b>	<b>0,41</b>	<b>0,405</b>
<b>5000000</b>	<b>2,03</b>	<b>2,76</b>
<b>10000000</b>	<b>4,05</b>	<b>6,285</b>
<b>15000000</b>	<b>6,75</b>	<b>10,15</b>
<b>20000000</b>	<b>8,13</b>	<b>14,67</b>

<i><b>Quick Sort First Pivot</b></i>	<b>N</b>	<b>Load(s)</b>	<b>t(s)</b>
	<b>1000</b>	<b>0</b>	<b>0,00</b>
	<b>10000</b>	<b>0</b>	<b>0,005</b>
	<b>50000</b>	<b>0,02</b>	<b>0,015</b>
	<b>100000</b>	<b>0,04</b>	<b>0,03</b>
	<b>150000</b>	<b>0,06</b>	<b>0,05</b>
	<b>200000</b>	<b>0,08</b>	<b>0,06</b>
	<b>250000</b>	<b>0,11</b>	<b>0,075</b>
	<b>500000</b>	<b>0,2</b>	<b>0,175</b>
	<b>1000000</b>	<b>0,4</b>	<b>0,4</b>
	<b>5000000</b>	<b>2,03</b>	<b>2,795</b>
	<b>10000000</b>	<b>4,06</b>	<b>6,36</b>
	<b>15000000</b>	<b>6,08</b>	<b>10,36</b>
	<b>20000000</b>	<b>8,14</b>	<b>14,545</b>

I tempi sono stati presi su due macchine diverse del laboratorio e poi è stata fatta una media aritmetica per ottenere un risultato più accurato.

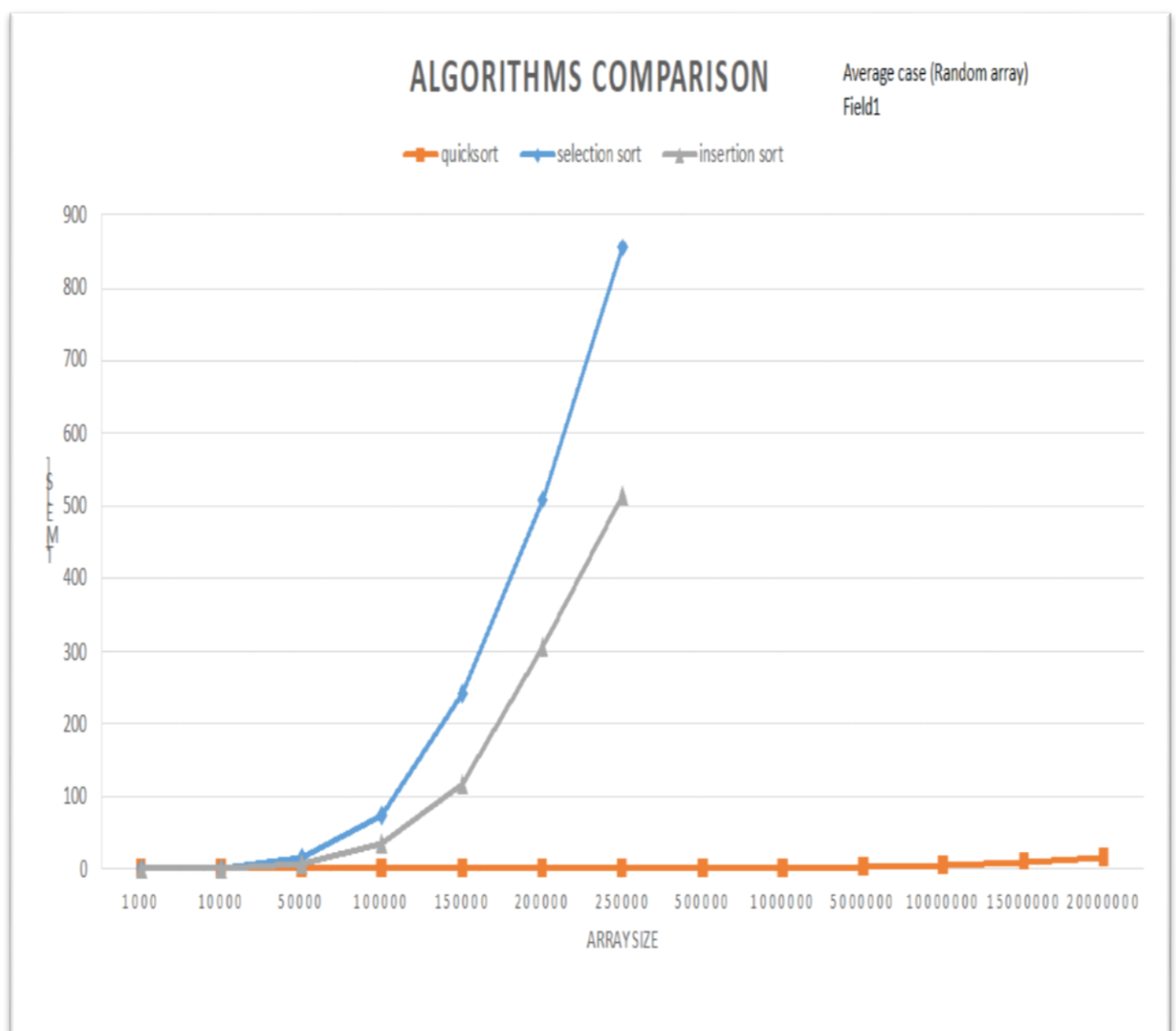
## CONCLUSIONI

Come ci aspettavamo l'algoritmo di ordinamento più veloce è stato il Quick Sort. Invece, l' Insertion Sort e il Selection Sort hanno fallito in quanto il loro

tempo medio di ordinamento superava i 10 minuti per un numero di record minore di 1 milione.

Non abbiamo notato grosse variazioni di tempi per quanto riguarda le tre implementazioni del Quick Sort nonostante scegliere il primo elemento come pivot porti al caso peggiore con costo  $\Theta(n^2)$ .

Viene allegato alla relazione un grafico che compara i tempi di ordinamento dei tre algoritmi e uno per i vari casi del Quick Sort.



Quicksort - pivot position

