

Università degli Studi di Torino

Relazione Laboratorio di Algoritmi

ESERCIZIO 1

Chironna Luca - Ferrero Regis Riccardo – Ferrod Roger

In questa breve relazione riportiamo i risultati ottenuti dall'esecuzione del Esercizio 1 confrontando le prestazioni dell'applicazione al variare di alcune specifiche degli algoritmi, in particolare sono stati presi in considerazione il numero di elementi da ordinare, la posizione del pivot dell'algoritmo Quicksort e i diversi tipi di dati. Lo scopo dell'esercizio è quello di implementare i seguenti tipi di algoritmi di ordinamento: Insertion Sort, Selection Sort, Quick Sort, per poi misurarne i tempi di risposta su un'applicazione che ne faceva uso.

INTRODUZIONE

L'applicativo agisce su un file contenente 20 milioni di records da ordinare composti dai seguenti campi:

- id: (tipo intero) identificatore univoco del record;
- field1: (tipo stringa) contiene parole estratte dalla divina commedia,
- field2: (tipo intero);
- field3: (tipo floating point);

Potete assumere che i valori non contengono spazi o virgole.

Ogni algoritmo è implementato in modo tale da poter essere utilizzato su un tipo generico e consente di specificare il criterio secondo cui ordinare i dati.

I tempi di risposta di questi algoritmi sono stati ricavati tramite la funzione `clock()` una funzione che ritorna in modo approssimato la CPU time che quindi differisce dal tempo reale di esecuzione osservabile dall'utente. Ogni valore raccolto è la media aritmetica di tre misurazioni effettuate sui computer del laboratorio aventi le seguenti caratteristiche:

- Scientific Linux 6.8(Carbon)
- 7.5 GB RAM
- 8 processori Intel Core i7-2600 CPU @ 3.40GHz

Per quanto visto a lezione ci aspettiamo che il Quicksort, avente complessità temporale media di $\Theta(n \log n)$ ¹, sia il più veloce dei tre siccome sia il Insertion Sort che il Selection Sort possiedono una complessità media di $\Theta(n^2)$. È bene notare che anche il Quicksort presenta una complessità di $\Theta(n^2)$ nel caso peggiore, per questo motivo abbiamo implementato il Quicksort con la possibilità di modificare la posizione del pivot che può quindi assumere un valore casuale (tra 0 e n-1), il valore mediano (n/2) oppure il primo e l'ultimo elemento dell'array.

RACCOLTA DATI

Le misurazione di seguito riportate si riferiscono esclusivamente al confronto del primo campo in quanto il confronto tra due stringhe di caratteri è più lento rispetto a due numeri.

La situazione che si presenta a seguito dell'avvenuto caricamento dell'array è sempre un caso medio poiché i valori dei tre campi sono distribuiti in maniera casuale rispetto all'ordine di lettura del file.

<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

¹ La variabile n indica la dimensione dell'array da ordinare.

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

<i>Quick Sort</i> <i>Median Pivot</i>	N	Load (s)	t (s)
	1000	0	0,00
	10000	0	0,01
	50000	0,025	0,015
	100000	0,045	0,025
	150000	0,06	0,05
	200000	0,085	0,06
	250000	0,11	0,075
	500000	0,21	0,165
	1000000	0,41	0,405
	5000000	2,03	2,76
	10000000	4,05	6,285
	15000000	6,75	10,15
	20000000	8,13	14,67

<i>Quick Sort</i> <i>First Pivot</i>	N	Load (s)	t (s)
	1000	0	0,00
	10000	0	0,005
	50000	0,02	0,015
	100000	0,04	0,03
	150000	0,06	0,05
	200000	0,08	0,06

250000	0,11	0,075
500000	0,2	0,175
1000000	0,4	0,4
5000000	2,03	2,795
10000000	4,06	6,36
15000000	6,08	10,36
20000000	8,14	14,545

ANALISI

Come ci aspettavamo l'algoritmo di ordinamento più veloce è stato il Quicksort, mentre l'Insertion Sort e il Selection Sort superano i 10 minuti già a partire da 1 milione di record. È possibile notare dalla Figura 1 come i tempi di risposta del Quicksort seguano un andamento logaritmico mentre Insertion Sort e Selection Sort presentano una curva parabolica che quindi rispecchia l'andamento quadratico descritto dalle formule teoriche di complessità.

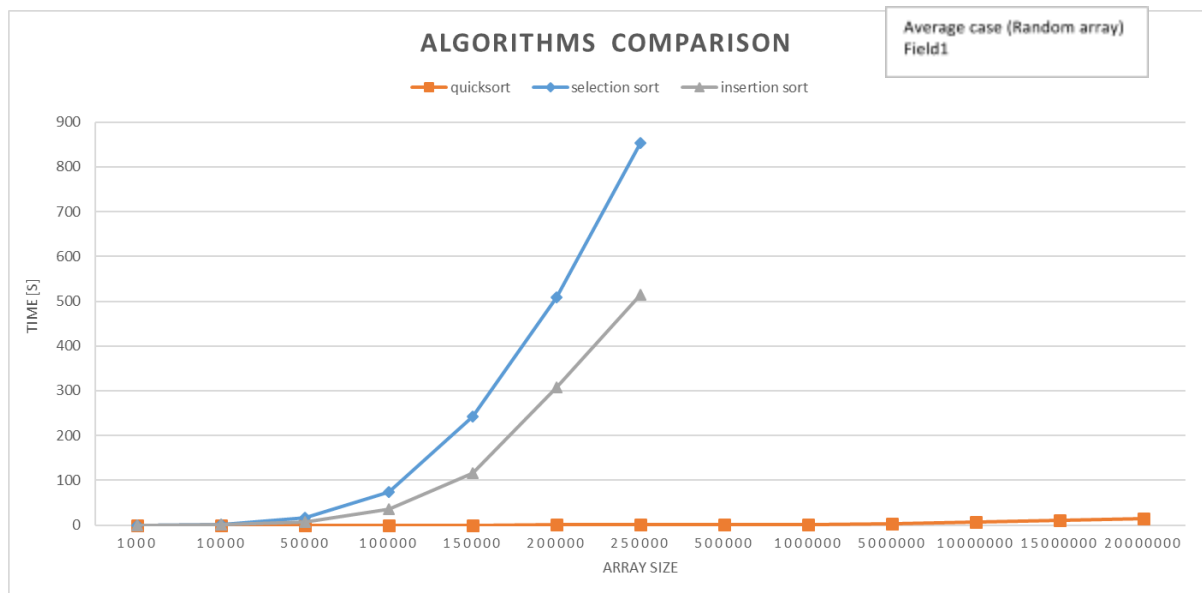


Figura 1

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 $\Theta(n^2)$.

