ALGORITMO PER RICERCA E CALCOLI SU TELEMETRIE

Progetto di Algoritmi e Strutture Dati

creato da

Stefano Zizzi

Università degli Studi di Urbino
Informatica Applicata

MATRICOLA: 312793

GIUGNO 2022

Indice

1 Specifica del problema

Si supponga di dover progettare un programma per la gestione della telemetria di un'auto da gara. Il sistema di telemetria permette di rilevare, con una frequenza di 10Hz (cioè 10 volte al secondo) i seguenti dati relativi al funzionamento dell'auto: il numero di giri del motore (in [*giri/minuto*]), la temperatura del motore (in [°C]), la temperatura del chip della centralina elettronica (in [°C]),la velocità dell'auto (in [km\h]), le accelerazioni rispetto a tre assi di riferimento x,y ,z (in [m\s^2]).
Si ipotizzi che le informazioni raccolte in un certo intervallo di tempo vengano poi scritte su un file in formato testo, secondo il seguente formato (si assumano campi separati da tabulazione o spazio):

- `Id:` un codice alfanumerico (concatenazione di 3 lettere e 3 numeri) che identifica univocamente la rilevazione in un dato istante.
- `GiriM:` un numero intero che rappresenta il numero di giri del motore nell'istante di rilevamento
- `TempM:` un numero reale che rappresenta la temperatura del motore nell'istante di rilevamento.
- `TempC:` un numero reale che rappresenta la temperatura del chip nell'istante di rilevamento.
- `Vel:` un numero reale che rappresenta la velocità dell'auto nell'istante di rilevamento.
- `AccX:` un numero reale che rappresenta l'accelerazione lungo l'asse x nell'istante di rilevamento.
- `AccY:` un numero reale che rappresenta l'accelerazione lungo l'asse y nell'istante di rilevamento.
- `AccZ:` un numero reale che rappresenta l'accelerazione lungo l'asse z nell'istante di rilevamento.

Ad esempio:

Id	GiriM	TempM	TempC	Vel	AccX	AccY:	AccZ
rba001	8760	110.1	45.3	186	8.2	0.1	1.0
qat121	6804	102.4	46.5	215	6.4	0.2	1.2
...	...	...	...	...	....	...	...
bmy247	10401	98.6	42.5	144	-3.3	0.1	-0.9
...	...	...	...	...	....	...	...

Scrivere un programma ANSI C che esegue le seguenti elaborazioni:

1. Acquisisce il file e memorizza le relative informazioni in una struttura dati dinamica di tipo albero.

2. Permette la ricerca dei dati relativi ad una data rilevazione.

3. Calcola valore medio, massimo e minimo di tutte le 7 grandezze fisiche (GiriM, TempM, TempC, Vel, AccX, AccY, AccZ) rispetto a tutte le rilevazioni.

Per quanto riguarda l'analisi teorica si deve fornire la complessità dell'algoritmo di ricerca, di quello del calcolo del valore medio, del minimo e del massimo. Oltre all'analisi teorica della complessità si deve effettuare uno studio sperimentale della stessa. Come suggerimento si può operare generando un numero N di rilevazioni casuali. L'analisi sperimentale deve quindi valutare la complessità al crescere di N per ognuna delle operazioni (ricerca, calcolo valore medio, massimo, minimo).

2 Analisi del Problema

2.1 Dati di Ingresso del Problema

I dati da fornire in ingresso al programma sono:

- Il nome del file di testo da cui estrapolare i dati delle telemetrie;
- L'Id dell'automobile di cui ricercare le telemetrie.

2.2 Dati di Uscita del Problema

I dati di uscita del programma sono:

- Le telemetrie dell'automobile associata all'Id ricercato;
- Valore Medio, Massimo e Minimo di tutte le 7 grandezze fisiche, rispetto a tutte le rilevazioni.

2.3 Relazioni Intercorrenti tra i Dati del Problema

L'Id associato all'automobile è univoco, dunque, non potranno esistere due automobili con lo stesso Id.
Le relazioni utilizzate sono quella di Massimo, Minimo e Media.
La media è la somma dei valori numerici divisa per il numero di valori numerici considerati. Per calcolare la media aritmetica tra due o più numeri basta sommarli e dividere il risultato ottenuto per il numero dei valori, dunque, possiamo rappresentarla come:

$$\frac{\sum_{i=1}^{n} telemetrie}{n}$$

Dove n è il numero di telemetrie contenute nel file di testo, e *telemetrie* rappresenta tutte le singole grandezze fisiche collegate ad ogni singola automobile.
Dato un insieme T, e un elemento $M \in T$, M è detto massimo assoluto se:

$$\forall\, x \in T \; M > x$$

Allo stesso modo dato un insieme T, e un elemento $m \in T$, x è detto minimo assoluto se:

$$\forall\, x \in T \; m < x$$

3 Progettazione dell'Algoritmo

3.1 Scelte di Progetto

La scelta del file di testo avviene direttamente nel momento dell'esecuzione, in quanto viene utilizzato il parametro *argv* della funzione main, ad esempio:

```
./prog dati.txt
```

Il progetto contiene due codici sorgente strutturati con delle librerie che contengono le funzioni principali: il primo finalizzato all'utente, il secondo finalizzato allo studio della complessità algoritmica (verranno indicati rispettivamente con (1) e (2)). Entrambi i codici hanno lo stesso funzionamento con la differenza che (1) contiene un menù più strutturato, e dà all'utente la possibilità di scegliere quali operazioni eseguire, mentre (2) presenta un menù più minimale e contiene un generatore di telemetrie, oltre che un contatore di istruzioni e un cronometro per il tempo di esecuzione.

All'esecuzione del programma, la versione (1) richiede all'utente di selezionare una tra le seguenti opzioni: Stampa degli Id Presenti, Ricerca di una Telemetria, Stampa di Massimo, Minimo e Media oppure Uscita dal programma. Alla fine di ognuna di queste viene data la possibilità di tornare al menù iniziale.
All'esecuzione della versione (2) viene chiesto all'utente il numero di elementi da generare, al fine di testare le prestazioni del programma.

Il file viene aperto in modalità lettura nella versione (1) mentre in modalità lettura e scrittura nella versione (2) , poichè richiede che il file venga riempito con i valori generati, se non presente, il file di testo verrà creato con il nome indicato negli argomenti del comando di esecuzione, mentre, nella versione (1), in caso il nome del file indicato fosse sbagliato, verrà segnalato un errore. Nel caso in cui, al momento dell'esecuzione, il file sia denominato in modo non adeguato, o se per altri motivi la lettura o scrittura del file non avvenga con successo, il programma termina senza eseguire calcoli.

I nodi dell'albero sono definiti tramite un puntatore ad un struttura a cui viene allocata la memoria necessaria (tramite le funzioni della libreria `stdlib.h`), contenente l'Id, le diverse grandezze fisiche e due puntatori, destro e sinistro, che determinano la struttura di albero binario. Al fine di ottenere un albero binario di ricerca è stato utilizzato un algoritmo di inserimento che basa il posizionamento dei nodi sul valore di ritorno della funzione `strcmp()` della libreria `string.h`, che ha come parametri l'Id del nodo da inserire e l'Id del nodo che viene fatto scorrere.

Al momento dell'inserimento non viene controllato se il formato dei contenuti del file di testo siano corretti o meno, in quanto secondo la specifica di progetto indica che le telemetrie vengono registrate in modo automatico. A seguito dell'inserimento le strutture che conterrano i risultati del Massimo e Minimo sono riempite con i valori della radice, in quanto una volta inizializzate i loro valori sono posti a 0, potenzialmente compromettendo il risultato finale.

La funzione di ricerca è la medesima nelle due versioni e sfrutta la funzione `strcmp()` per determinare se il nodo attraversato è quello contenente le telemetrie ricercate. In caso l'Id non sia formulato nel modo corretto (controllo effettuato tramite l'utilizzo di funzioni della libreria `ctype.h`) o non sia presente all'interno dell'albero, il programma richiede all'utente di inserire un Id valido. Per facilitare l'utilizzo del programma, nella versione (1) viene data la possibilità all'utente di stampare gli Id presenti da poi poter ricercare.

Il calcolo di Massimo, Minimo e Media viene effettuato all'interno della stessa funzione di visita in ordine anticipato dell'albero, al fine di diminuire il numero di visite complessivo. Nella visita in ordine anticipato, prima si elabora il valore contenuto nel nodo al quale si e giunti, poi si visitano con lo stesso algoritmo il sottoalbero sinistro e il sottoalbero destro del nodo I rispettivi risultati vengono inseriti in strutture diverse per poi essere stampati. Il calcolo della media avviene sommando tutti i valori rispettivi alle stesse grandezze fisiche e dividendoli per il numero di elementi direttamente nel momento della stampa
Nella versione (1) la visita contiene anche un contatore del numero di nodi inseriti, diversamente da (2) che li calcola all'inserimento; la scelta è determinata dalla differenza di opzioni date all'utente nelle due versioni.

La stampa dei risultati avviene tramite la stessa funzione a cui viene passata la struttura, e un parametro che seleziona se si debba dividere i dati da stampare per il numero di nodi o meno (per la stampa della media).

Nella versione (2) vengono anche stampati il tempo di esecuzione [*cps*] e il numero di istruzioni eseguite, in particolare, quante istruzioni hanno richiesto Inserimento, Ricerca e Calcolo di Massimo, Minimo e Media.

Al termine del programma viene liberata la memoria allocata alle diverse strutture.

Nella versione (1) per il menù utente è stata utilizzata una funzione di pulizia schermo, per rendere l'esperienza migliore. Al fine di rendere la funzione portabile è stato utilizzato il seguente comando nella fase di inclusione librerie:

```
#ifdef _WIN32
#include <conio.h>
#else
#include <stdio.h>
#define clrscr() printf("\e[1;1H\e[2J")
#endif
```

In seguito viene riportato il menù utente della versione (1):

```
    ===================================
    =              MENU                 =
    ===================================

    Scegliere tra le seguenti opzioni:
    1. Stampa telemetrie
    2. Cerca telemetria
    3. Stampa Minimo,Massimo e Media
    0. Esci
```

3.2 Passi dell'Algoritmo

I seguenti passi riguardano la versione di codice finalizzata all'utente:

1. Acquisire dati dal file scelto dall'utente.
2. Inserire dati estrapolati da file in un albero binario di ricerca.
3. Calcolare valori Massimi, Minimi e Medi.
4. In base alla scelta fatta dall'utente:
    - Stampare tutti gli Id Presenti;
    - Ricercare una Telemetria;
    - Stampare i valori Massimi, Minimi e Medi.

Per evitare ridondanze di codice in tutte le fasi dell'algoritmo sono stati utilizzati dei sottoprogrammi.

4 Implementazione Algoritmo

4.1 Funzioni principali

Data la scelta di non inserire il codice per intero nella relazione, spiegherò brevemente il ruolo delle diverse funzioni:
Contenuti del file `main.c`:

- `int menu()`: la funzione stampa il menù per l'utente e restituisce il valore relativo alla sua scelta.
- `void elabora_file()`: la funzione prende come parametri il puntatore al file, un nodo di input e la radice, scorre il file e richiama la funzione di inserimento nell albero per ogni riga del file, saltando la prima, poichè contiene gli indici dei dati.
- `int controlla_id()`: la funzione prende come parametro una stringa da controllare, affinchè rispetti il formato alfanumerico dell'Id(3 lettere e 3 numeri), restituisce 1 se la stringa non rispetta i parametri, oppure 0 se la stringa è accettabile.
- `void ricerca()`: la funzione prende come parametri la radice e il nodo in cui memorizzare i dati risultanti dalla ricerca. Alla chiamata la funzione richiede l'Id dell'automobile da ricercare, viene controllato e ricercato tramite una chiamata alla funzione di ricerca nell'albero.
- `void stampa_struttura()`: la funzione prende come parametri la struttura da stampare, un intero che è utilizzato per selezionare se la struttura da stampare sia la media o meno, e il numero di nodi dell'albero.

Contenuti del file `lib.c`:

- `int inserisci_in_albero_bin_ric()`: la funzione prende come parametri un nodo che contiene i valori da inserire nell'albero e la radice. É la funzione che rende l'albero binario, un albero di ricerca, poichè inserisce i parametri in base al risultato della funzione `strcmp()`. La funzione restituisce 1 se l'inserimento è avvenuto con successo e 0 nel caso contrario.
- `void visita_albero_bin_ant()`: la funzione prende come parametri la radice e il numero di nodi attraversati. Il compito della funzione è quello di stampare gli Id dei nodi presenti nell'albero in colonne di 10 elementi.
- `nodo_albero_bin_t* cerca_in_albero_bin_ric()`: la funzione prende come parametri la radice e l'Id da ricercare,restituisce l'indirizzo del nodo che contiene il valore se presente,`NULL` se il valore è assente.
- `void visita_albero_bin_ant_calcoli()`: la funzione prende come parametri la radice, tre diversi nodi per i risultati dei Massimi, Minimi e Media e un contatore del numero di nodi. Tramite una serie di istruzioni `if`, calcola il Massimo e il Minimo, mentre nella struttura Media inserisce la somma di tutte le grandezze fisiche singolarmente.

5 Testing del Programma

Qui sono riportati una serie di test significativi:

5.1 Test 1

```
Inserire ID automobile da ricercare, in formato alfanumerico (abc123),
   oppure scrivere 'esci' per uscire:
=> zre106

=======================
Telemetrie automobile:
=======================
GiriM: 6601
TempM: 91.83
TempC: 41.67
Vel: 198.80
AccX: -3.20
AccY: 0.42
AccZ: -1.90
Premi invio per tornare al menu.
--------------------------------------------------------------------------------
Massimi:
=======================
GiriM: 19008
TempM: 108.43
TempC: 49.81
Vel: 213.28
AccX: 6.56
AccY: 0.74
AccZ: 1.36

=======================
Minimi:
=======================
GiriM: 2061
TempM: 91.30
TempC: 40.12
Vel: 136.05
AccX: -9.90
AccY: -0.73
AccZ: -1.90

=======================
Media:
=======================
GiriM: 12481
TempM: 100.30
TempC: 45.48
Vel: 186.87
AccX: -1.66
AccY: 0.10
AccZ: -0.03


Premi invio per tornare al menu.
```

5.2 Test 2

```
Inserire ID automobile da ricercare, in formato alfanumerico (abc123),
    oppure scrivere 'esci' per uscire:
=> ctr272

========================
Telemetrie automobile:
========================
GiriM: 5552
TempM: 106.35
TempC: 44.04
Vel: 174.58
AccX: -8.40
AccY: 0.65
AccZ: 0.98
Premi invio per tornare al menu.
--------------------------------------------------------------------------------
========================
Massimi:
========================
GiriM: 19687
TempM: 109.65
TempC: 49.21
Vel: 205.14
AccX: 8.53
AccY: 0.98
AccZ: 1.74

========================
Minimi:
========================
GiriM: 1496
TempM: 90.08
TempC: 40.65
Vel: 131.67
AccX: -8.40
AccY: 0.17
AccZ: -1.70

========================
Media:
========================
GiriM: 6225
TempM: 99.01
TempC: 44.33
Vel: 163.08
AccX: -1.92
AccY: 0.53
AccZ: 0.09


Premi invio per tornare al menu.
```

5.3 Test 3

```
Inserire ID automobile da ricercare, in formato alfanumerico (abc123),
    oppure scrivere 'esci' per uscire:
=> qhl684

========================
Telemetrie automobile:
========================
GiriM: 4849
TempM: 100.88
TempC: 41.36
Vel: 148.98
AccX: 5.89
AccY: 0.88
AccZ: 1.86
Premi invio per tornare al menu.
--------------------------------------------------------------------------------
========================
Massimi:
========================
GiriM: 15138
TempM: 108.01
TempC: 49.69
Vel: 209.73
AccX: 9.64
AccY: 0.88
AccZ: 1.86

========================
Minimi:
========================
GiriM: 2057
TempM: 90.66
TempC: 41.18
Vel: 132.89
AccX: -7.30
AccY: 0.05
AccZ: -1.50

========================
Media:
========================
GiriM: 7267
TempM: 99.63
TempC: 45.64
Vel: 161.41
AccX: 2.67
AccY: 0.41
AccZ: 0.22


Premi invio per tornare al menu.
```

5.4 Test 4

```
Inserire ID automobile da ricercare, in formato alfanumerico (abc123),
    oppure scrivere 'esci' per uscire:
=> sti847

========================
Telemetrie automobile:
========================
GiriM: 16458
TempM: 102.88
TempC: 42.27
Vel: 157.71
AccX: -2.10
AccY: 0.47
AccZ: -1.70
Premi invio per tornare al menu.

--------------------------------------------------------------------------------
========================
Massimi:
========================
GiriM: 17946
TempM: 109.94
TempC: 48.95
Vel: 219.23
AccX: 8.04
AccY: 0.98
AccZ: 1.61

========================
Minimi:
========================
GiriM: 3971
TempM: 90.12
TempC: 40.75
Vel: 144.37
AccX: -8.70
AccY: 0.31
AccZ: -1.70

========================
Media:
========================
GiriM: 10795
TempM: 100.76
TempC: 44.71
Vel: 184.57
AccX: -2.19
AccY: 0.57
AccZ: -0.33


Premi invio per tornare al menu.
```

5.5 Test 5

```
Inserire ID automobile da ricercare, in formato alfanumerico (abc123),
    oppure scrivere 'esci' per uscire:
=> wzm477

========================
Telemetrie automobile:
========================
GiriM: 19626
TempM: 105.95
TempC: 43.80
Vel: 216.89
AccX: 0.89
AccY: 0.12
AccZ: 0.87
Premi invio per tornare al menu.

--------------------------------------------------------------------------------

========================
Massimi:
========================
GiriM: 19626
TempM: 109.32
TempC: 48.60
Vel: 216.89
AccX: 7.53
AccY: 0.94
AccZ: 1.52

========================
Minimi:
========================
GiriM: 4736
TempM: 92.30
TempC: 40.40
Vel: 154.45
AccX: -9.50
AccY: 0.10
AccZ: -1.90

========================
Media:
========================
GiriM: 13383
TempM: 101.42
TempC: 44.03
Vel: 182.88
AccX: -0.56
AccY: 0.40
AccZ: -0.03


Premi invio per tornare al menu.
```

5.6 Test 6

```
Inserire ID automobile da ricercare, in formato alfanumerico (abc123),
    oppure scrivere 'esci' per uscire:
=> xjn666

========================
Telemetrie automobile:
========================
GiriM: 7008
TempM: 102.20
TempC: 41.96
Vel: 200.12
AccX: -2.70
AccY: 0.44
AccZ: -1.80
Premi invio per tornare al menu.

--------------------------------------------------------------------------------
========================
Massimi:
========================
GiriM: 19434
TempM: 108.88
TempC: 48.30
Vel: 214.41
AccX: 7.00
AccY: 0.76
AccZ: 1.44

========================
Minimi:
========================
GiriM: 2836
TempM: 91.78
TempC: 40.04
Vel: 137.73
AccX: -9.90
AccY: 0.08
AccZ: -1.90

========================
Media:
========================
GiriM: 9878
TempM: 98.90
TempC: 42.78
Vel: 171.61
AccX: -2.05
AccY: 0.42
AccZ: -0.12


Premi invio per tornare al menu.
```

6 Analisi Teorica della Complessità

Poichè il maggiore contributo alla complessità è dato dalle funzioni di Inserimento, Ricerca e Visita, saranno le uniche soggette allo studio della complessità.

6.1 Complessità Inserimento e Ricerca

La complessita asintotica dei due algoritmi dipende dal numero di iterazioni delle istruzioni `for`. Nel seguito indichiamo con n il numero di nodi dell'albero binario di ricerca e con h la sua altezza. Quando l'inserimento avviene in un'albero vuoto o quando il valore da ricercare è già contenuto nella radice abbiamo il caso migliore, che sarebbe

$$T(0) = 1$$

Nel caso pessimo il numero di iterazioni è proporzionale ad h, in caso particolare se l'albero è degenerato in una lista otteniamo il caso pessimo.

$$T(n) = n$$

Nel caso medio il numero di iterazioni è proporzionale alla lunghezza media del percorso per raggiungere un nodo dalla radice, che è dimostrabile sia:

$$T(n) = O(logn)$$

6.2 Complessità Massimi, Minimi e Media

Trattandosi comunque di un'algoritmo di visita, è prevedibile che non otterremo meno di n operazioni , chiaramente quando l'albero è vuoto facciamo una sola operazione quindi la complessità sarà:

$$T(0) = 1$$

A prescindere dai casi facciamo un'operazione inerente alla condizione `if`, i calcoli di Massimo, Minimo hanno stesso identico comportamento, in quanto a prescindere dal risultato, ad ogni visita compiono 7 controlli con l'istruzione `if`, più altri 7 assegnamenti ciascuno, se i controlli risultano positivi. In modo leggermente diverso il calcolo della Media richiede sempre 7 assegnamenti per ogni elemento.
Questo ci lascia con:

$$T(n) = 1 + 7 + 7 + 7 + 7 + 7 + T(k) + T(n - 1 - k)$$

dove k è il numero di nodi del sottoalbero sinistro del nodo al quale si e giunti ed $n-k-1$ è il numero di nodi del suo sottoalbero destro. Sommando otteniamo:

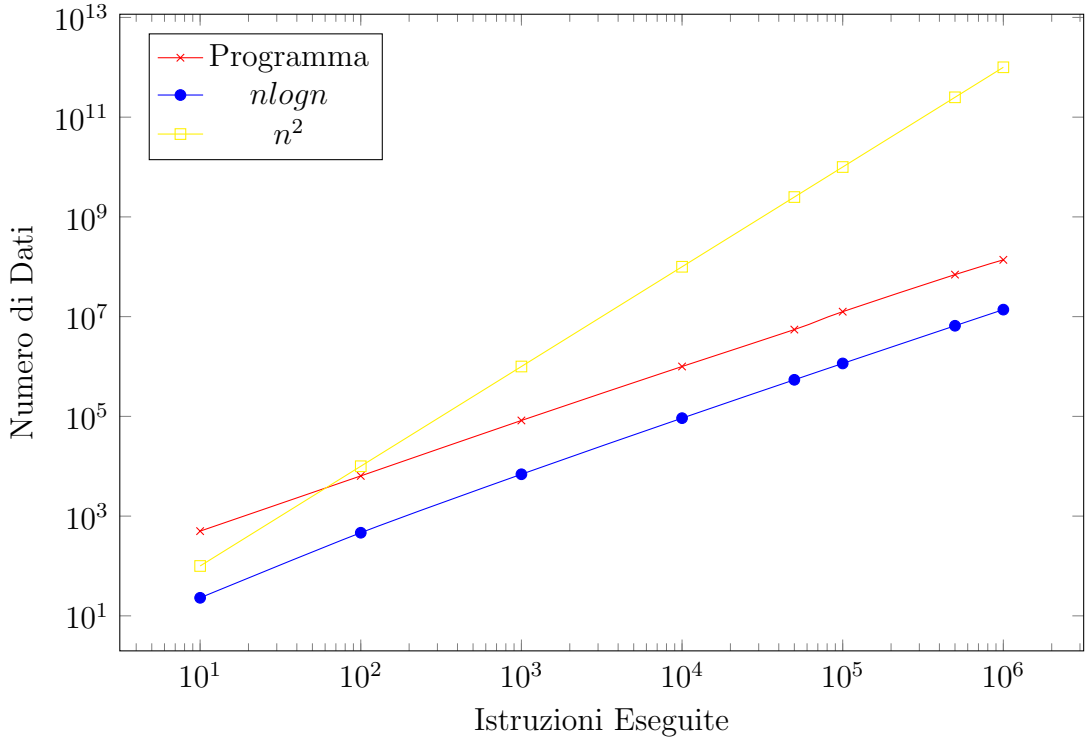
$$T(n) = 36 + T(k) + T(n - 1 - k)$$

Che per induzione è dimostrabile sia $T(n) = O(n)$.

7 Analisi Sperimentale

Nonostante la specifica di progetto richiedesse di studiare solamente la complessità degli algoritmi di Ricerca e calcolo di Massimo, Minimo e Media quì sono riportati ulteriori risultati per rendere più preciso lo studio complessivo del programma.

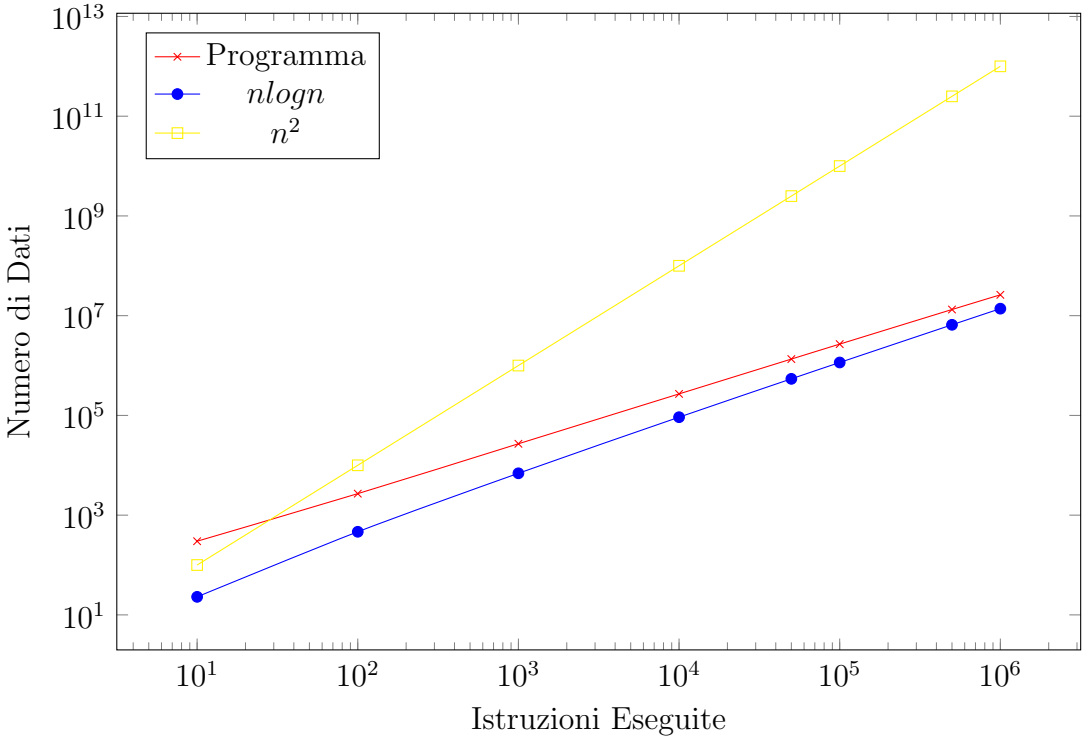
Quì è riportato un grafico che mostra l'andamento del programma considerando anche l'algoritmo di inserimento a confronto con $nlogn$ e n^2:



Come si può notare il programma ha un'andamento pari a $T(n) = O(10nlogn)$ che può essere assimilato a $T(n) = O(nlogn)$, dunque pseudolineare.
Nella seguente tabella è riportato l'impatto percentuale delle singole funzioni sul numero totale di istruzioni, all'aumentare del numero di dati:

	10	100	1000	10000	50000	100000	500000	1000000
Inserimento	46.75%	62.03%	69.30%	75.65%	77.34%	78.68%	80.75%	82.23%
Ricerca	4.87%	0.45%	0.00%	00.00%	00.00%	00.00%	00.00%	00.00%
Massimi	17.37%	12.71%	10.25%	8.12%	7.55%	7.15%	6.42%	5.92%
Minimi	16.23%	12.57%	10.24%	8.12%	7.55%	7.15%	6.42%	5.92%
Media	14.77%	12.24%	10.20%	8.12%	7.55%	7.15%	6.42%	5.92%

Quì è riportato il grafico dell'andamento del programma senza considerare l'Inserimento:



Quì è riportato un grafico con le funzioni messe a confronto. Massimo, Minimo e Media sono stati messi sotto lo stesso nome di "Visita", in quanto è stato riscontrato lo stesso identico andamento.