Analisi empirica degli algoritmi di ordinamento

Graziano Francesco [1] , Ongaro Michele [2] , Petri Riccardo [3] e Ungaro Marco [4]

Università degli Studi di Udine, Dipartimento di Matematica e Informatica

A.A. 2024-2025

[1]Email: graziano.francesco@spes.uniud.it, Matricola: 166680
[2]Email: ongaro.michele@spes.uniud.it, Matricola: 168049
[3]Email: petri.riccardo@spes.uniud.it, Matricola: 167623
[4]Email: ungaro.marco@spes.uniud.it, Matricola: 168934

Contenuti

1 Introduzione **2**

2 Counting Sort **3**

3 Quick Sort **4**

4 Quick Sort 3 way **5**

5 Radix Sort **6**

6 Conclusioni **7**

Capitolo 1

Introduzione

Il progetto richiede l'implementazione di quattro algoritmi di ordinamento per array interi di dimensioni variabili. Gli algoritmi che andremo ad analizzare sono il Counting Sort, il Quick Sort, il Quick Sort 3 way e il Radix Sort (algoritmo a scelta). Oltre alla corretta implementazione viene richiesto di effettuare un analisi empirica dei tempi medi di esecuzione degli algorimti al variare della dimensione dell'array e del range dei valori interi. Per stimare i tempi di esecuzione di questi algoritmi garanantendo un errore relativo massimo pari a 0.001 adotteremo le seguenti metodologie:

- Utilizzeremo un clock di sistema monotono per garantire precisione nelle misurazioni (ad esempio, `perf_counter()` del modulo *time* in Python);
- Andremo a generare almeno 100 campioni per ciascun grafico, con i valori dei parametri (dimensione dell'array n e intervallo dei valori m) distribuiti secondo una progressione geometrica;
- Effettueremo più esecuzioni per ogni campione, per stimare in modo affidabile il tempo medio di esecuzione e, eventualmente, il relativo errore.

Dopo aver stimato i tempi di esecuzione per ciascun algoritmo, risulterà interessante confrontare i grafici ottenuti per analizzare il comportamento degli algoritmi in diverse situazioni, come il caso peggiore, quello migliore o in quello medio.

Da tutto questo potremmo ottenere una verifica empirica dell'andamento asintotico dei tempi di esecuzione di ogni algoritmo.

Capitolo 2

Counting Sort

Spiegazione dell'algoritmo di ordinamento
- Analisi della complessità
- Grafico dei tempi di esecuzione

Capitolo 3

Quick Sort

Capitolo 4

Quick Sort 3 way

Capitolo 5

Radix Sort

Capitolo 6

Conclusioni