



Esercizio numero 1

Tiziano Jhonny Floriddia , 1041726

Nicolas Fiocco, 1052526

Giorgio Francesco Floridia , 1054584

10/12/2024

Relazione sui Test di Merge Sort e Quick Sort

OBIETTIVO

L'obiettivo del lavoro è stato confrontare le prestazioni degli algoritmi di ordinamento *Merge Sort* e *Quick Sort*, implementati in linguaggio C, su un dataset di grandi dimensioni contenente 20 milioni di record. Ogni record era composto da tre campi: una stringa, un numero intero e un numero in virgola mobile.

METODOLOGIA

I test hanno misurato i tempi di esecuzione per ciascun algoritmo utilizzando ognuno dei tre campi come chiave di ordinamento. L'esecuzione veniva interrotta se un algoritmo impiegava più di 10 minuti per completare un ordinamento. L'analisi dei risultati si è focalizzata sui tempi di risposta e sulla stabilità degli algoritmi. Il tempo di esecuzione è la somma del tempo di lettura + tempo di ordinamento + tempo di scrittura.

RISULTATI

ALGORITMO	TIPO DI DATO	TEMPO DI ESECUZIONE (s)
Merge Sort	Stringa	116,07
	Intero	103,32
	Float	105,18
Quick Sort	Stringa	//
	Intero	98,56
	Float	99,78

CONSIDERAZIONI

1. Tempi di Esecuzione

I risultati mostrano che il *Merge Sort* è generalmente stabile ed efficiente su tutti i tipi di dati, rispettando la sua complessità media $O(n \log n)$. Tuttavia, i tempi per le stringhe sono superiori rispetto agli altri tipi di dati.

Il *Quick Sort* si è dimostrato più rapido per i dati numerici, ma ha fallito l'ordinamento per stringhe di grandi dimensioni.

2. Confronto tra Algoritmi

Il *Quick Sort* è mediamente più veloce del *Merge Sort* per numeri interi e numeri in virgola mobile, mentre il *Merge Sort* ha il vantaggio di essere stabile e privo di errori anche con input più complessi.

3. Fallimento dell'Operazione

L'implementazione del *Quick Sort* ha incontrato difficoltà significative con la presenza di molte stringhe duplicate, dove l'elevato numero di confronti ha causato il rallentamento massivo dell'algoritmo. Una possibile soluzione potrebbe essere una migliore scelta del pivot.

4. Distribuzione dei Dati

Entrambi gli algoritmi mostrano prestazioni influenzate non solo dal tipo di dato, ma anche dalla distribuzione e dalle caratteristiche dei dati stessi. Ad esempio, i record con molteplici valori duplicati hanno impattato negativamente sul *Quick Sort*.