

Relazione Laboratorio di Algoritmi

Membri del gruppo: Mateusz Lubiejewski, Joana Minja, Saimir Pasho

Sorting

Il calcolo delle tempistiche è comprensivo della lettura da file record.csv ma non comprende le varie stampe di verifica presenti durante i test.

Quick Sort :

- Impiega 1 minuto e 28 secondi per ordinare 1 milione di record
- 2 secondi quando i record sono 100 mila
- dai 2 milioni in su, si presenta un errore di StackOverflow

Merge Sort:

- Impiega 9 secondi per ordinare 1 milione di record
- 28 secondi per 2 milioni
- 1 secondo per 100 mila
- e fallisce nell'ordinamento di 20 milioni di record

Insertion Sort:

- Impiega 6 minuti e 55 secondi per ordinare 100 mila record
- In tutti gli altri casi si verifica fallimento perché supera i 10 minuti nell'esecuzione

Heap Sort:

- Impiega 9 secondi per ordinare 1 milione di record
- 26 secondi per 2 milioni di record
- 1 secondo per 100 mila

Come ci aspettavamo l'HeapSort e il MergeSort si sono dimostrati gli algoritmi migliori, perchè hanno la complessità $O(n \log n)$, mentre la complessità dell'InsertionSort e del QuickSort è $O(n^2)$. Tuttavia, il QuickSort nel caso medio e nel caso migliore è pseudolineare, invece l'InsertionSort anche nel caso medio è quadratico e solo nel caso migliore (quando i dati sono già ordinati o quasi) è lineare.

Non è mai stato possibile ordinare tutti i 20 milioni di record perché già durante la lettura da file si verificava un errore di Out of Memory.

Mateusz Lubiejewski

Hash

Il calcolo delle tempistiche è comprensivo della lettura da file record.csv ma non comprende le varie stampe di verifica presenti durante i test.

Gli inserimenti di qualsiasi tipo su 2 milioni di record hanno una durata di 12 secondi, indipendentemente se si tratti dell'inserimento utilizzando come chiave il primo, il secondo o il terzo elemento del record.

Le find allo stesso modo impiegano tutte 11 secondi su 1 milione di record, indipendentemente del tipo usato come chiave.

Il tempo impiegato per la cancellazione di 1 milione di record con qualsiasi chiave è sempre di 11 secondi.

BinaryTree

Il calcolo delle tempistiche è comprensivo della lettura da file record.csv ma non comprende le varie stampe di verifica presenti durante i test.

Il tempo dell' inserimento degli elementi di tipo Number di 30 mila record è di 22 secondi, dai 40 mila compreso, in poi l'inserimento fallisce a causa di un errore StackOverflow.

Inserimento di tipo Double con 1 milione di record viene eseguita in 14 secondi, invece quella di String con la stessa dimensione viene eseguita in 4 minuti e 26 secondi.

Il tempo di esecuzione della find cambia sempre di molto, come nell'inserimento, al cambiamento del tipo cercato. Per il Number si verifica sempre StackOverflow con record maggiori o uguali a 40 mila, invece 30 mila vengono trovati in 28 secondi.

La ricerca con il tipo Double impiega 18 secondi su 1 milione; sullo stesso numero di record, la String impiega 4 minuti e 34 secondi.

La cancellazione di Number su 30 mila record è di 20 secondi, su 1 milione invece per Double e String è rispettivamente di 22 secondi e 5 minuti e 36 secondi.

Notiamo quindi che la differenza sostanziale tra i due contenitori è che l'hash non cambia comportamento con il cambiare del tipo dell'elemento, invece il binary search muta ogni volta. Inoltre come possiamo notare, nel primo contenitore, i tempi di inserimento, ricerca e cancellazione sono pressoché gli stessi; nel secondo nessun tempo è uguale.

Ci aspettavamo delle differenze di comportamento tra i due contenitori e soprattutto che l'hash fosse più veloce e più costante nelle tempistiche.