Relazione Finale – Algoritmi e Strutture Dati

- Lorenzo Corrias 66247

Considerazioni sugli Algoritmi implementati

Gli algoritmi analizzati nell'ultima esercitazioni di laboratorio e di cui si riportano qui i dati raccolti hanno seguito, prevedibilmente, le nozioni imparate durante gli studi teorici. La tabella contenente i tempi di esecuzione e il numero di ordinamenti eseguiti ci permette di dedurre che l'esecuzione di ogni algoritmo di sorting varia, talvolta anche notevolmente, le sue prestazioni relativamente al tipo di ordinamento dell'insieme di partenza fornito. Il QuickSort ha, per esempio, riportato dei tempi di esecuzione molto diversi tra il suo caso ottimo (insiemi casuali e quasi ordinati, per i quali si considera una complessità computazionale linearitmica) e il suo caso pessimo (insieme inversamente ordinato, con complessità quadratica).

Algoritmi ottimi e Tempi di Esecuzione

Gli algoritmi con prestazioni migliori si sono confermati essere l'HeapSort e il MergeSort, entrambi di complessità linearitmica. Sebbene il tempo di calcolo sia un fattore determinante nella scelta di implementazione di un algoritmo di sorting, è tuttavia bene ricordare che questi metodi di ordinamento consentono sì un'esecuzione rapida, ma allo stesso tempo richiedono un uso della memoria abbastanza significativo, trattandosi entrambi di algoritmi ricorsivi. In generale, inoltre, la scelta di utilizzare di un algoritmo di sorting rispetto ad un altro deve anche essere presa tenendo in considerazione la tipologia e l'omogeneità dei dati in ingresso, poiché, come abbiamo riscontrato empiricamente, ogni algoritmo possiede dei casi migliori e peggiori per la sua esecuzione. Basti notare, per esempio, come nel caso di un array già ordinato l'algoritmo di ordinamento per inserzione si sia dimostrato migliore di tutti gli altri.

Conclusioni e Considerazioni Finali

I dati forniti all'interno della tabella sono ovviamente influenzati anche da fattori esterni, come la potenza della macchina su cui il programma di calcolo viene avviato e il carico della stessa. Tuttavia, analizzando non le singole statistiche ma gli andamenti generali, possiamo concludere che l'implementazione di algoritmi ottimi, come risultano essere il MergeSort e l'HeapSort, siano estremamente convenienti per casi in cui si hanno a disposizione moli di dati estese e molto eterogenee, per le quali l'utilizzo massiccio della memoria del calcolatore può essere un fattore da prendere in considerazione in cambio di una estrema rapidità di esecuzione. D'altro canto, possiamo concludere che per insiemi di dimensioni modeste, nella scelta dell'algoritmo deve essere determinante un'analisi accurata del tipo di dati che si vogliono confrontare, ma in generale algoritmi meno veloci e performanti risultano comunque preferibili, dato che consentono un'implementazione più facile e rapida.