Algoritmi di ordinamento e i loro casi di uso

Guillen, Youssef

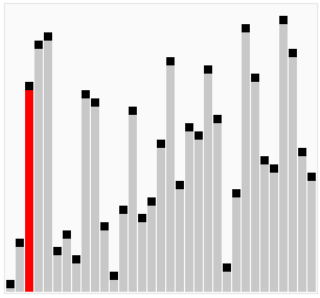
March 25, 2025

Introduzione:

In questo elaborato verrà illustrata la modalità d'uso dei vari algoritmi di ordinamento; con annessa misurazione del tempo che impiega un algoritmo ad ordinare N elementi.

1 bubble sort:

Il Bubble Sort è un algoritmo di ordinamento semplice e intuitivo che funziona confrontando coppie adiacenti di elementi in una lista e scambiandoli se sono nell'ordine sbagliato. Questo processo continua ripetutamente fino a quando l'intera lista è ordinata. L'algoritmo "fa salire" gli elementi più grandi (o più piccoli, a seconda dell'ordinamento desiderato) verso la fine della lista. Il nome "Bubble" (bolla) deriva dal fatto che le coppie di elementi vengono inmagazzinati in delle bolle e poi scambiati se necessario. Il funzionamento del Bubble Sort si articola in più passaggi. In ogni passaggio, l'algoritmo confronta coppie di elementi adiacenti e li scambia se il primo è maggiore (nel caso di ordinamento crescente) del secondo. Dopo ogni iterazione, l'elemento più grande è posizionato nella sua posizione corretta, quindi la parte della lista che deve essere ulteriormente esaminata diminuisce progressivamente. Il processo continua fino a quando non sono più necessari scambi, indicando che la lista è ordinata.



1.1 complessità:

La complessità temporale del Bubble Sort è $O(n^2)$ nel caso medio e nel caso peggiore. Questo perché, nel caso di una lista non ordinata, l'algoritmo effettua $n-1$ confronti nel primo passaggio, $n-2$ nel secondo, e così via. Di conseguenza, il numero totale di confronti è proporzionale a n^2, il che rende l'algoritmo inefficiente per liste di grandi dimensioni. Tuttavia, nel caso migliore, quando la lista è già ordinata, l'algoritmo può essere ottimizzato per terminare subito dopo una singola scansione, riducendo la complessità a $O(n)$. Il Bubble Sort viene spesso

utilizzato in contesti educativi per insegnare i concetti di base degli algoritmi di ordinamento, grazie alla sua semplicità. Tuttavia, a causa della sua inefficienza, non è consigliato per l'ordinamento di grandi dataset.

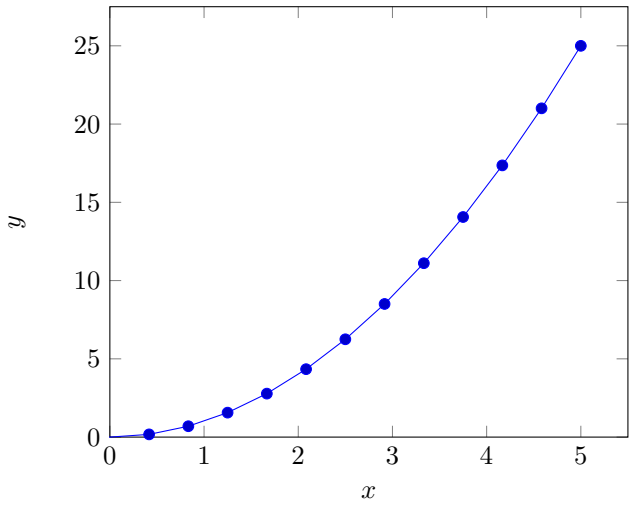
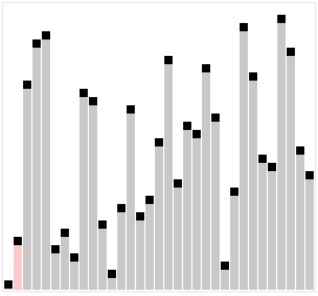


Figure 1: grafico dell'equazione della complessita del algoritmo bubble sort

2 Selection Sort:



Selection Sort è un algoritmo di ordinamento semplice che funziona selezionando ripetutamente l'elemento minimo (o massimo, a seconda del caso) dalla parte non ordinata della lista e spostandolo alla fine della parte ordinata. L'algoritmo suddivide la lista in due sezioni: una ordinata e una non ordinata. Inizialmente, la parte ordinata è vuota, mentre la parte non ordinata contiene tutti gli elementi. Ogni iterazione dell'algoritmo consiste nell'individuare il minimo (o massimo) tra gli elementi non ordinati e spostarlo alla fine della parte ordinata. La procedura continua fino a che tutti gli elementi sono stati ordinati, e la parte non ordinata è vuota. La semplicità dell'algoritmo è la sua caratteristica distintiva, ed è per questo che viene spesso usato in ambito didattico per spiegare i concetti di base degli algoritmi di ordinamento. Tuttavia, nonostante la sua semplicità, Selection Sort ha dei limiti importanti in termini di efficienza, specialmente quando la dimensione dei dati aumenta.

2.1 complessità:

La complessità temporale del Selection Sort è $O(n^2)$, poiché, per ogni elemento della lista, l'algoritmo deve eseguire una scansione completa della parte non ordinata per trovare l'elemento minimo. Questo porta a un numero di confronti che cresce quadraticamente con il numero di elementi: per il primo elemento si effettuano $n-1$ confronti, per il secondo $n-2$, e così via, fino all'ultimo. Quindi, anche nel caso migliore, quando la lista è già ordinata, il numero di confronti rimane $O(n^2)$.

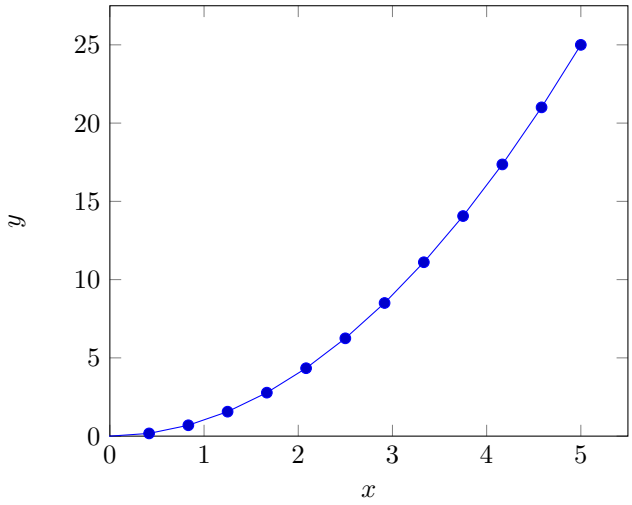
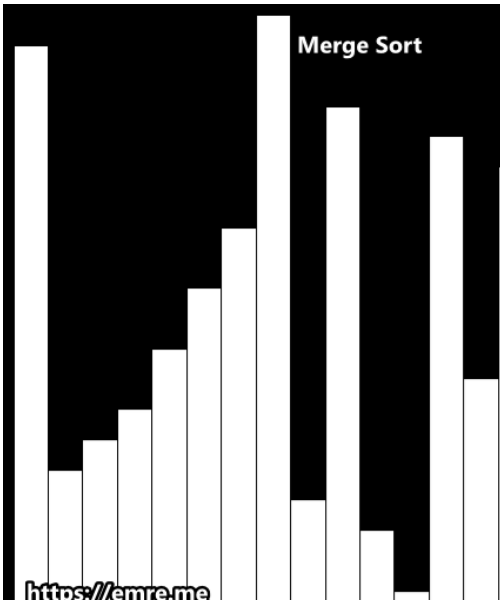


Figure 2: grafico dell'equazione della complessita del algoritmo selection sort

3 Merge sort:

Il Merge Sort è un algoritmo di ordinamento basato sul principio "divide et impera". Divide ricorsivamente la lista in due metà, le ordina separatamente e poi le unisce (merge) in un'operazione di ordinamento. La sua complessità temporale è $O(nlogn)$ nel caso medio e nel peggiore, il che lo rende molto più efficiente rispetto ai precedenti algoritmi per dataset di grandi dimensioni. È spesso utilizzato in contesti di programmazione avanzata e applicazioni in cui la stabilità dell'ordinamento è critica.



4 Heap sort:

5 Insertion sort:

L'Insertion Sort è un algoritmo che costruisce l'ordinamento finale della lista uno elemento alla volta. Funziona suddividendo la lista in una parte ordinata e una parte non ordinata, e inserendo ripetutamente gli elementi dalla parte non ordinata nella posizione corretta della parte ordinata. Ha una complessità temporale di $O(n^2)$ nel caso medio, ma può essere molto efficiente per liste già parzialmente ordinate. È frequentemente utilizzato in situazioni in cui i dati sono già quasi ordinati o quando le liste sono piccole.