

1.1 Concetti base dell'Informatica: Algoritmi

Insegnamento di Informatica

Elisabetta Ronchieri

Corso di Laurea di Economia, Università di Ferrara

I semestre, anno 2014-2015



Argomenti

Algoritmi notevoli

Algoritmi di ordinamento



Argomenti

Algoritmi notevoli

Algoritmi di ordinamento



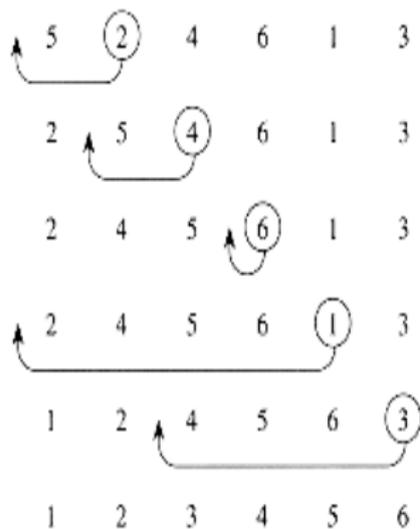
Algoritmo di ordinamento

- ▶ Altre classi:
 - ▶ Insertion Sort;
 - ▶ Shell Sort;
 - ▶ Quick Sort;
 - ▶ Merge Sort.



Insertion Sort

- ▶ É un algoritmo particolarmente semplice.
- ▶ É un algoritmo di inserimento.
- ▶ Ordina i primi due elementi dell'insieme.
- ▶ Inserisce il terzo elemento nella posizione corretta rispetto ai primi due.
- ▶ Continua con la procedura fino all'inserimento di tutti gli elementi, e quindi all'ordinamento.



Insertion Sort

- ▶ Rispetto al Selection Sort o Bubble Sort, il numero di confronti che effettua é legato all'ordinamento iniziale dell'insieme.
 - ▶ Nei casi peggiori, non é migliore del Bubble Sort e del Selection Sort.
 - ▶ Nei casi medi, é meno lento.
- ▶ Se l'insieme é ordinato, i confronti saranno $n - 1$ con n numero di elementi dell'insieme.
- ▶ Altrimenti sono dell'ordine di n^2 .
- ▶ Garantisce un numero basso di confronti, ma effettua un numero elevato di spostamenti.
- ▶ Ha senso usarlo per insiemi quasi ordinati.

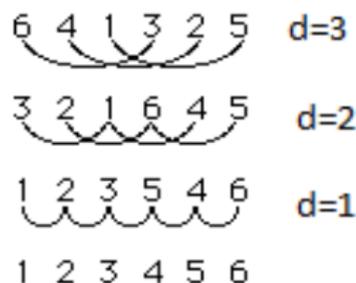


Shell Sort

- ▶ É un algoritmo abbastanza evoluto.
- ▶ É un Algoritmo di inserimento.
- ▶ Ordina cercando di ridurre gli inserimenti.
- ▶ Confronta tutti gli elementi che si trovano ad una distanza d , scambiandoli se non sono ordinati.
- ▶ Riduce di volta in volta il valore di d .
- ▶ Quando $d = 1$, gli elementi sono adiacenti e l'algoritmo si interrompe.

Esempio:

su $n = 6$, $d = 3$ alla prima iterazione.



Shell Sort

- ▶ Non é facile valutare il valore ottimale d .
- ▶ In genere:
 - ▶ si ordina l'insieme con una spaziatura grande;
 - ▶ si riduce la spaziatura durante l'iterazione;
 - ▶ si riordina l'insieme.
- ▶ Shell Sort é migliore del Bubble Sort, Insertion Sort e Selection Sort.
- ▶ Ha la peculiaritá di scambiare velocemente gli elementi.



Quick Sort

- ▶ É un algoritmo evoluto.
- ▶ É un algoritmo di scambio e ricorsivo.
- ▶ L'idea:
 - ▶ dato un insieme di n elementi, si divide in due sottoinsiemi di $\frac{n}{2}$ elementi ciascuno, che si ordinano separatamente;
 - ▶ riunendo le due sottoparti già ordinate, si dimezza il tempo di calcolo;
 - ▶ l'insieme iniziale può essere suddiviso in un numero di elementi inferiore a $\frac{n}{2}$ per dimezzare ulteriormente il tempo di calcolo.
- ▶ Suddivide l'insieme in due sezioni e identifica un pivot nell'insieme.
- ▶ Gli elementi $> o =$ al pivot vanno in una parte, mentre quelli minori nell'altra.
- ▶ Itera il procedimento per ognuna delle sezioni rimanenti fino ad avere l'insieme ordinato, ogni volta identificando un pivot per ogni sezione.



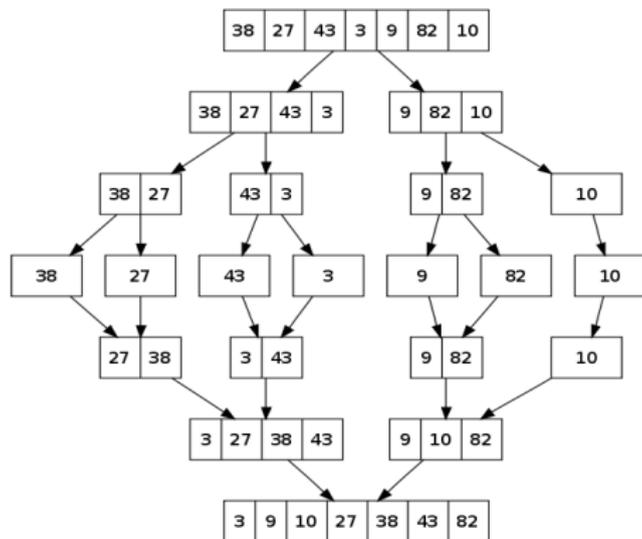
Quick Sort

- ▶ Caso migliore:
 - ▶ Il pivot divide l'insieme in due parti uguali ad ogni passo.
- ▶ Caso peggiore:
 - ▶ Il pivot divide l'insieme degli elementi in due parti: una composta da un solo elemento e l'altra dagli $n - 1$ restanti.
- ▶ Caso medio:
 - ▶ Tutti gli elementi possono essere presi come pivot con probabilità $\frac{1}{n}$.
 - ▶ La generica suddivisione $i, n - i$ avviene con probabilità $\frac{1}{n}$.



Merge Sort

- ▶ É un algoritmo evoluto.
- ▶ É un algoritmo ricorsivo.
- ▶ Sfrutta il concetto di fusione di insiemi ordinati.
- ▶ Divide l'insieme in due parti che ordina ricorsivamente.
- ▶ Unisce le sequenze ordinate risultanti.
- ▶ Si esce dalla ricorsione quando gli insiemi hanno una dimensione pari a 1.



Provenienza della figura:

http://www.jefclaes.be/2011/07/merge-sorting-in-javascript_1458.html



Confronto tra algoritmi di ordinamento

Algoritmi con soluzioni sempre piú sofisticate hanno complessit  computazionale sempre piú bassa.

Complessit  computazionale   basata sul numero di operazioni elementari necessarie ad effettuare l'ordinamento.

Algoritmo	Caso peggiore	Caso medio	Caso migliore
Insertion Sort	n^2	-	n
Bubble Sort	n^2	-	n
Selection Sort	n^2	-	n^2
Quick Sort	n^2	$n \cdot \ln(n)$	$n \cdot \log_2(n)$
Merge Sort	$n \cdot \log_2(n)$	$n \cdot \log_2(n)$	$n \cdot \log_2(n)$

