

Università degli Studi di Trieste

Corsi di ingegneria industriale e navale

Esercitazioni di Fondamenti di Informatica

Giacomo Strangolino

mailto/chat: delleceste@gmail.com

Materiale didattico anche su:

<http://www.giacomos.it>

(<http://www.giacomos.it/didattica/units/2011/>)

Lezione 5 (23/11/2011)

- Scrivere una funzione che esegua l'ordinamento di un vettore con l'algoritmo *bubble sort*
 - saranno confrontati il primo e il secondo elemento, poi il secondo e il terzo, poi il terzo e il quarto, e così via fino al confronto fra il penultimo e l'ultimo elemento.
 - Ad ogni confronto, se i due elementi confrontati non sono ordinati essi vengono scambiati.
 - Durante ogni iterazione almeno un valore viene spostato rapidamente fino a raggiungere la sua collocazione definitiva; in particolare, alla prima iterazione il numero più grande raggiunge l'ultima posizione del vettore

Lezione 5

- Il bubble sort effettua all'incirca $N^{**2}/2$ confronti e $N^{**2}/2$ scambi sia in media che nel caso peggiore.

NOTA: L'operatore `**` in python è l'elevamento a potenza

Lezione 5

Ottimizzazioni:

- se, in una data iterazione, non avviene alcuno scambio, allora il vettore è necessariamente ordinato e l'algoritmo può essere terminato anticipatamente (homework);
- se una data iterazione non sposta nessun elemento di posizione maggiore di un dato valore i , allora si può facilmente dimostrare che nessuna iterazione successiva eseguirà alcuno scambio in posizioni successive a tale valore i . L'algoritmo può dunque essere ottimizzato memorizzando l'indice a cui avviene l'ultimo scambio durante una iterazione, e limitando le iterazioni successive alla scansione del vettore solo fino a tale posizione.

Lezione 5

Ricerca binaria all'interno di un vettore ordinato

- Scrivere una funzione che, dato in ingresso un vettore *ordinato* e un numero a piacere, dica se il numero è contenuto nel vettore oppure no

Lezione 5

Ricerca binaria all'interno di un vettore ordinato

- l'idea è quella di iniziare la ricerca dall'elemento centrale, cioè a metà del vettore (si pensi alla ricerca di una parola in un vocabolario).
- Si **confronta** questo elemento con quello cercato:
- se **corrisponde**, la ricerca termina con successo;
- se è **inferiore**, la ricerca viene ripetuta sugli elementi precedenti (ovvero sulla prima metà del vettore), scartando quelli successivi;
- se invece è **superiore**, la ricerca viene ripetuta sugli elementi successivi (ovvero sulla seconda metà del vettore), scartando quelli precedenti.
- Quando tutti gli elementi sono stati scartati, la ricerca termina indicando che il valore non è stato trovato.

Lezione 5

Ricerca binaria all'interno di un vettore ordinato

- **Tempo di esecuzione $\text{ceil}(\log_2(N))$ (quindi più veloce di una ricerca lineare)**
 - × A causa del tipo di accesso casuale, può risultare non adatto se i dati sono in una *cache*;
 - × Per ricerca su memorie lente (disco), conviene abbandonare la ricerca binaria in favore di quella lineare quando l'intervallo si riduce a 16/8 elementi;
 - × Per ricerche multiple, a N costante, si selezionerà sempre $N/2$ per primo e poi $N/4$ o $3*N/4$ e così via: potrebbe essere utile memorizzare localmente questi valori in una memoria veloce se i dati risiedono in una memoria lenta

Lezione 5

Homework

- Scrivere una funzione atta a calcolare le radici di un'equazione di secondo grado
- Scrivere una funzione che, data una stringa in ingresso, conti il numero di linee, parole, caratteri letti.
- Scrivere una funzione *checkdate(g, m, a)* che controlla la validità della data passata come parametro.
- Scrivere una funzione che inverte gli elementi di una lista di interi. Si usi la funzione python *reverse()* su una lista come esempio e test della funzione scritta.
- Scrivere una funzione per riempire una matrice NxN a spirale, con valori crescenti da 0 a $N^2 - 1$ iniziando dall'angolo superiore sinistro. La funzione riceve come parametro N.

Lezione 5

Homework

Esempio matrice a spirale 5x5

0	1	2	3	4
15	16	17	18	5
14	23	24	19	6
13	22	21	20	7
12	11	10	9	8