Programmazione 1 - Modulo C

Ordinamento di Array

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

Dicembre 2020

Concetti generali

- Con il termine ordinamento si intende indicare il processo di disposizione di un insieme di oggetti secondo un ordine specifico;
- Scopo dell'ordinamento é quello di facilitare successive ricerche dei membri dell'insieme che é stato ordinato;
- Si tratta di un'attivitá fondamentale, che viene eseguita quasi universalmente:
  - elenchi telefonici;
  - archivi;
  - indici, biblioteche, dizionari, magazzini, ...

Ordinamento di array

Obiettivo principale:

- Efficienza in memoria: le permutazione (operazione fondamentale per mettere gli elementi in ordine) deve essere eseguita sul posto;
- Efficienza di tempo: minimizzare il numero di confronti da fare;

Metodi di ordinamento

- Metodi diretti:
  - tecniche semplici;
  - facili da capire e da realizzare;
  - complessitá dell'ordine di n^2;
- Metodi avanzati:
  - piú complessi e meno intuitivi;
  - A differenza dei metodi diretti, che spostano un elemento di una posizione ad ogni passo elementare, tali metodi si basano sul principio di spostare gli elementi per distanze maggiori con un unico salto;
  - Richiedono, generalmente, un numero di confronti nell'ordine di $n \log n$

Metodi di diretti

- Ordinamento per scambi: *Bubble sort*

- Ordinamento per inserimento: *Insertion sort*

Bubble sort

- Richiede il confronto e lo swap di elementi successivi, iterando sui dati
- Il metodo consiste nel confrontare elementi vicini tra loro portando l'elemento maggiore in ultima posizione durante la prima "passata" (eseguendo n-1 confronti), poi il secondo massimo in penultima posizione e cosí via finché l'array non é ordinato.
- Complessitá nel caso peggiore: $O(n^2)$

Bubble sort

Array da ordinare →

9	8	4	5	6	2	3	7	1	0

Ciclo # 1:

coppia	relazione	scambio									
1	9>8	a_1<->a_2	8	9	4	5	6	2	3	7	1	0

coppia	relazione	scambio									
2	9>4	a_2<->a_3	8	4	9	5	6	2	3	7	1	0

.....

coppia	relazione	scambio									
9	9>0	a_9<->a_{10}	8	4	5	6	2	3	7	1	0	9

Bubble sort

Ciclo # 2:

coppia	relazione	scambio									
1	8>4	a_1<->a_2	4	8	5	6	2	3	7	1	0	9

coppia	relazione	scambio									
8	8>0	a_8<->a_9	4	5	6	2	3	7	1	0	8	9

Ciclo # 3:

coppia	relazione	scambio									
3	6>2	a_3<->a_4	4	5	2	6	3	7	1	0	8	9

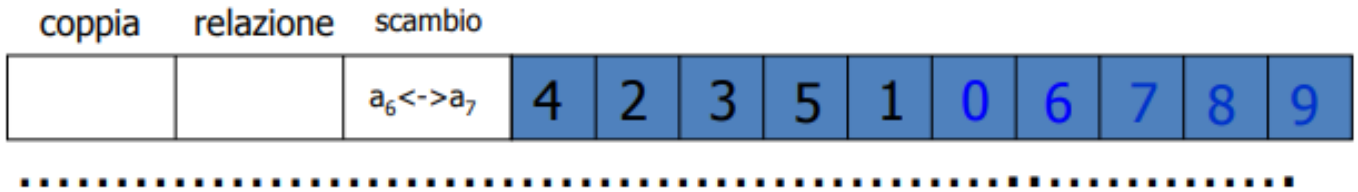
coppia	relazione	scambio									
4	6>3	a_4<->a_5	4	5	2	3	6	7	1	0	8	9

coppia	relazione	scambio									
6	7>1	a_6<->a_7	4	5	2	3	6	1	7	0	8	9

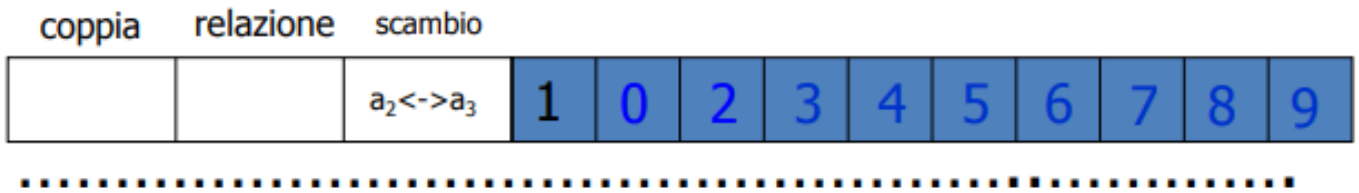
coppia	relazione	scambio									
7	7>0	a_7<->a_8	4	5	2	3	6	1	0	7	8	9

Bubble sort

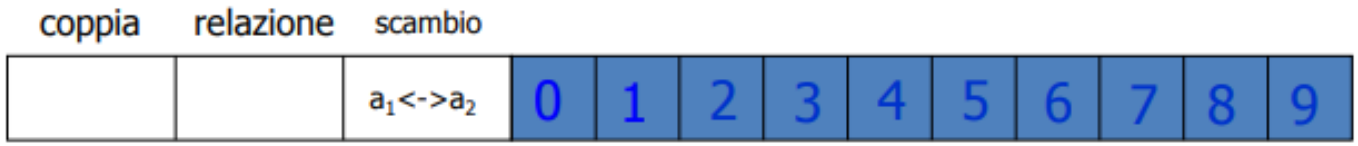
Ciclo # 4, al termine:



Ciclo # 8, al termine:



Ciclo # 9



Bubble sort in Java

Loop esterno sugli elementi:

```
47  public static void ordina(int[] a){
48  /**
49  Precondizione: a != null
50  Postcondizione: il vettore dati è ordinato in ordine crescente
51  */
52      for (int i = a.length - 1; i > 0; i--)
53              bolli(a,i);
54      }
55
```

Bubble sort in Java

Loop interno sugli elementi (modella lo scorrimento dell'elemento piú grande lungo il vettore fin a raggiungere la parte destra come una bolla di aria che sale dal basso verso l'alto in un bicchiere):

```
public static void bolli(int [] a, int i) {
/**
Precondizione: 0 ≤ i < dati.length
Postcondizione: la posizione i del vettore dati contiene l'elemento massimo del
sottovettore di dati compreso tra le posizioni 0 ed i
*/

    for (int j = 0; j < i; j++)
        if (a[j] > a[j+1])
            scambia(a,j,j+1);
    }
```

Bubble sort in Java

Scambia gli elementi:

```java
public static void scambia(int[] a, int i, int j) {
    int x;
    x = a[i];
    a[i] = a[j];
    a[j] = x;
}
```

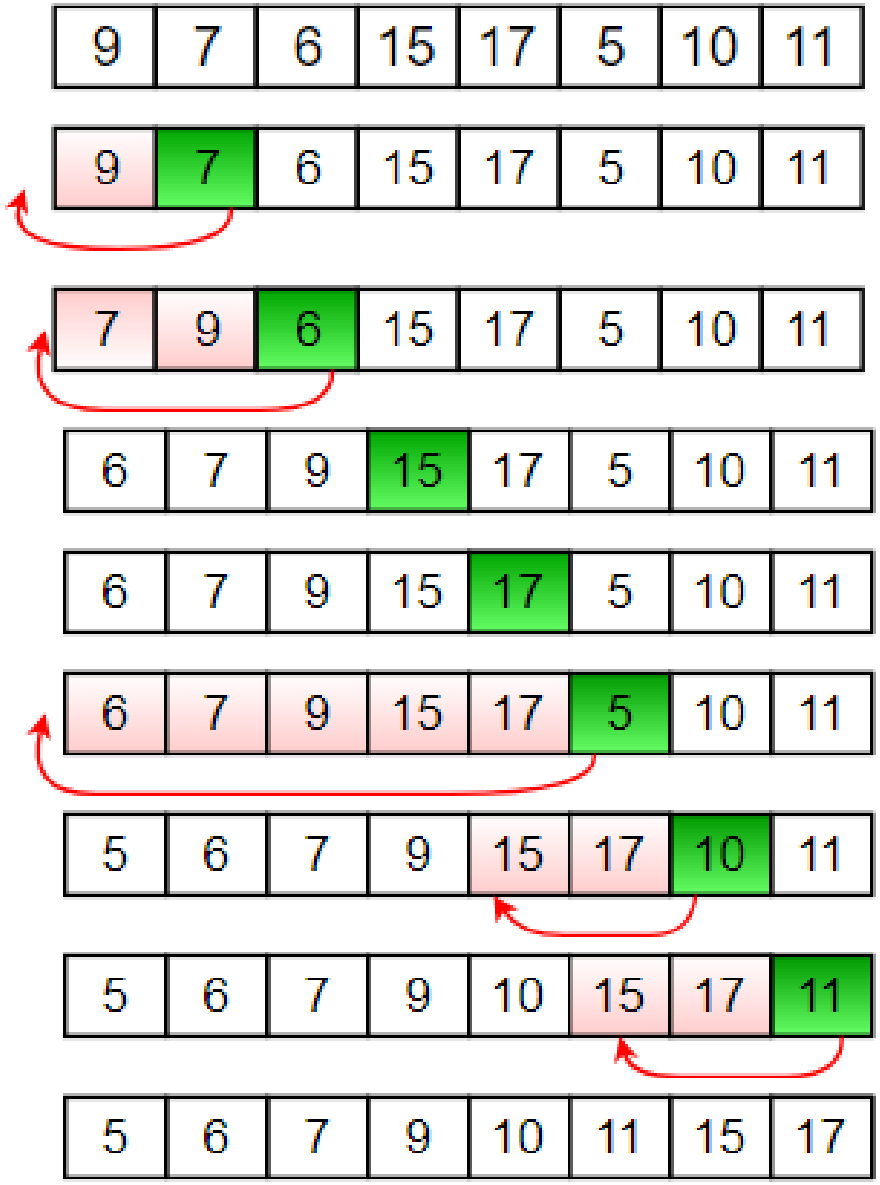
Insertion sort

Funziona allo stesso modo usato da molte persone per ordinare le carte da tenere in mano.

1. Si inizia con la mano sinistra vuota e le carte coperte sul tavolo
2. Si prende dal tavolo una carta e la si inserisce nella corretta posizione nella mano sinistra. Per trovare la posizione corretta si confronta la nuova carta con ogni altra carta giá nella mano, scorrendole tutte una alla volta.
3. Si ripete il passo 2 sino a quando il mazzo sul tavolo diventa vuoto.

Complessitá nel caso peggiore: $O(n^2)$

Insertion sort



Insertion sort in Java

```java
public static void insertionSort(int[] a) {
        int j,v;
        for (int i = 1; i < a.length; i++) {
            v = a[i];
            j = i;
            while (j > 0 && a[j-1] > v) {
                a[j] = a[j-1];
                j--;
            }
            a[j] = v;
        }
    }
```