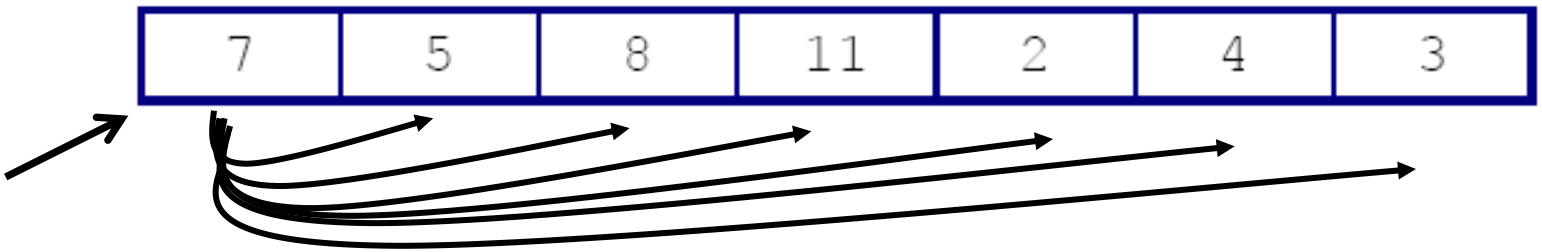
Algoritmi di Ordinamento

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

Ordinamento per sostituzione

Per implementare l'Algoritmo si devono usare 2 indici :

- Uno (I) che tiene conto della posizione in cui si trova l'elemento da ordinare (primo, secondo, terzo, …)
- Uno (J) che permette di scorrere l'array per effettuare il confronto ed eventualmente lo scambio



Ordinamento per sostituzione

I passi da seguire sono i seguenti :

1) Posizionamento sul primo elemento dell'array
2) Confronto dell'elemento con tutti gli elementi successivi e scambio ogni volta che se ne trova uno più piccolo
3) Posizionamento sul secondo elemento dell'array
4) Confronto dell'elemento con tutti gli elementi successivi e scambio ogni volta che se ne trova uno più piccolo
5) Posizionamento sul terzo elemento dell'array
6) Confronto dell'elemento con tutti gli elementi successivi e scambio ogni volta che se ne trova uno più piccolo
7) Tale procedimento viene ripetuto N-1 volte

Osservazione : Per implementare l'Algoritmo abbiamo bisogno di 2 indici :

- Uno che tiene conto della posizione in cui si trova l'elemento considerato (primo, secondo, terzo, …)
- Uno che permette di scorrere la parte successiva dell'array per effetuare i confronti e gli eventuali scambi

Ordinamento per sostituzione

Esempio : dato l'array :

7	5	8	11	2	4	3

i passi, per un ordinamento crescente, sono i seguenti :

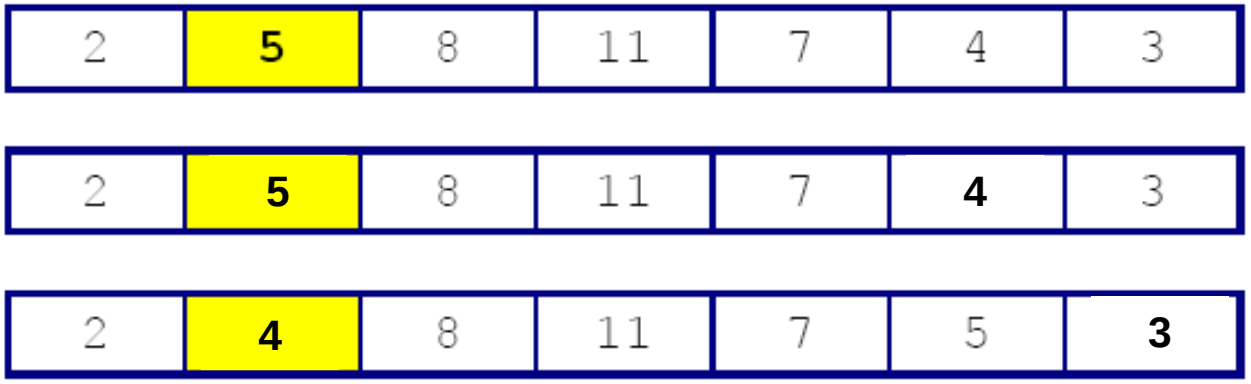
1^ posizione

7	5	8	11	2	4	3

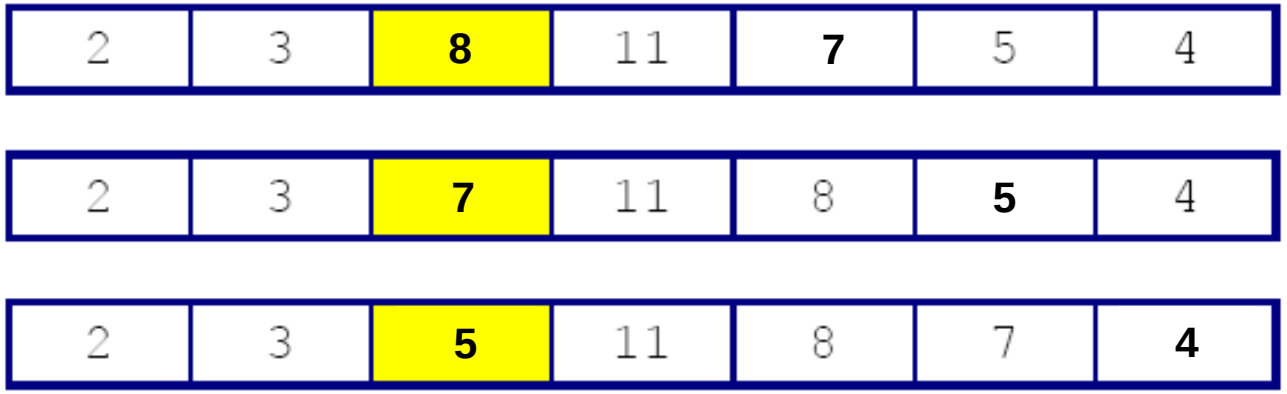
5	7	8	11	2	4	3

Ordinamento per sostituzione

2^ posizione

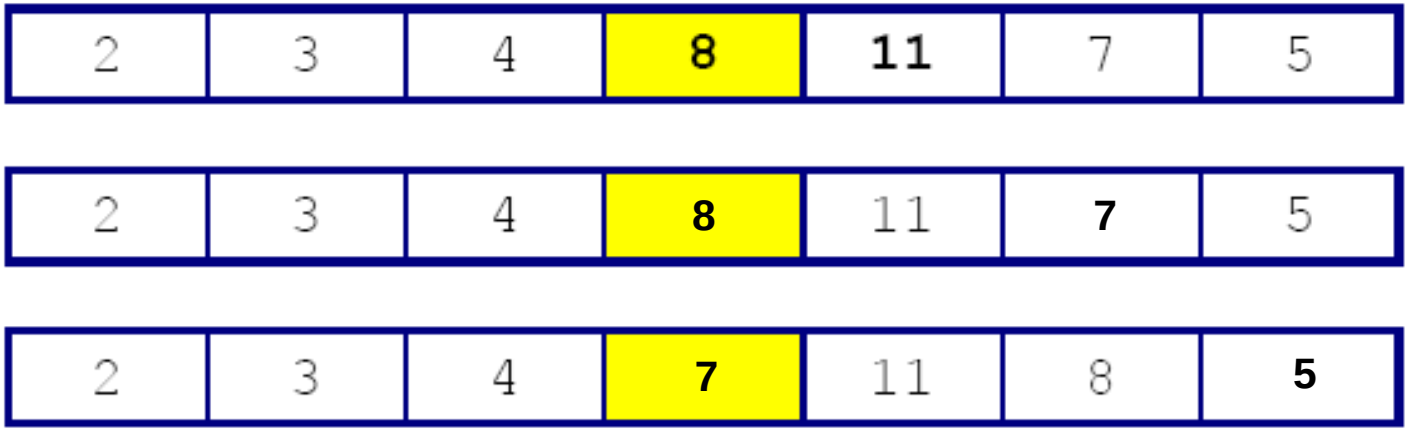


3^ posizione

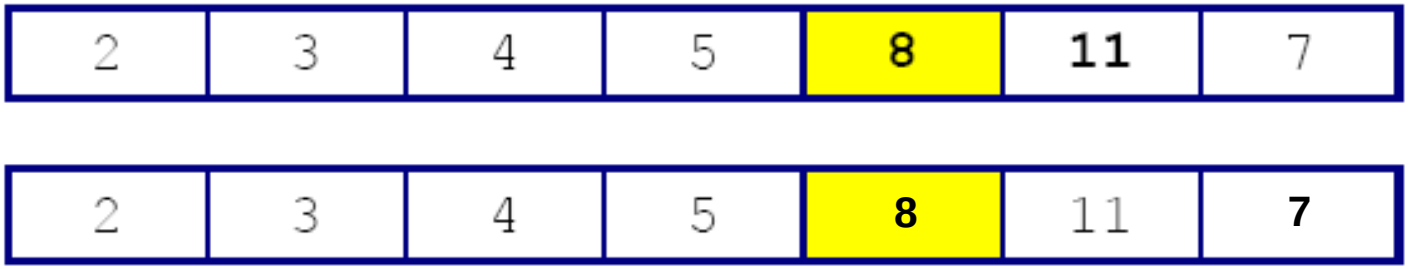


Ordinamento per sostituzione

4^ posizione



5^ posizione



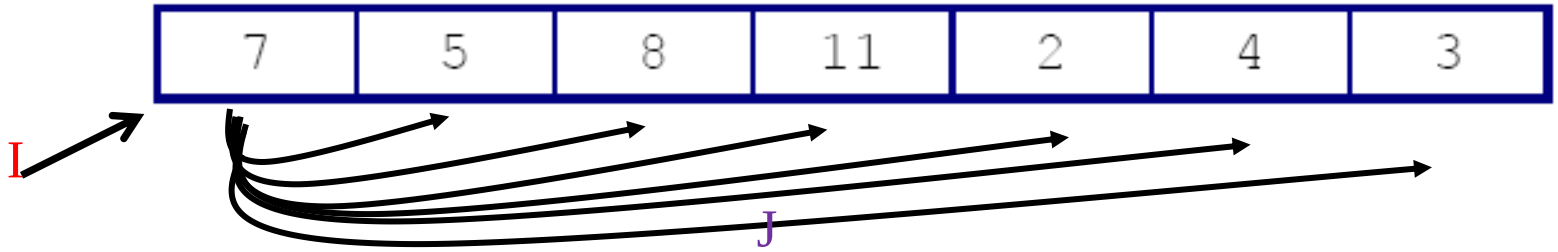
Ordinamento per sostituzione

6^ e 7^ posizione

2	3	4	5	7	**11**	**8**

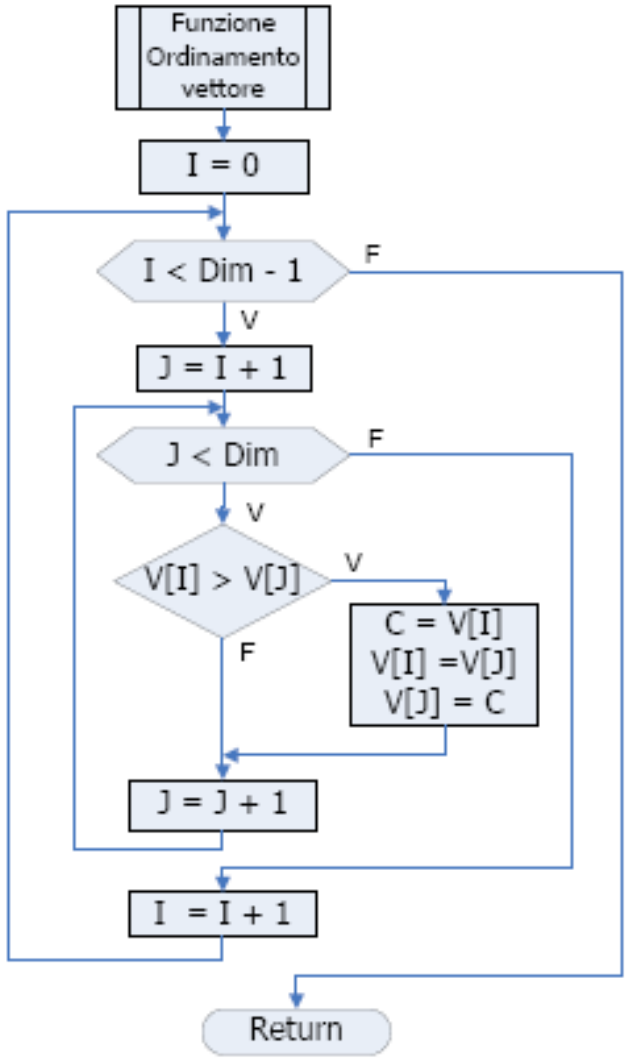
Come abbiamo già detto per implementare l'Algoritmo si devono usare 2 indici :

- Uno (I) che tiene conto della posizione in cui si trova l'elemento da ordinare (primo, secondo, terzo, …)
- Uno (J) che permette di scorrere l'array per effettuare il confronto ed eventualmente lo scambio



Flowchart e Codifica in C

```
void Ordina(void){
Int I,J,C;
I=0;
while(I<Dim-1)
{
  J=I+1;
  while(J<Dim)
   {
    if(V[I]>V[J])
      {
        C=V[I];
        V[I]=V[J];
        V[J]=C;
       }
    J=J+1
    }
 I=I+1;
}
}
```



Bubble-sort (ordinamento a bolle)

L'ordinamento a bolle è un algoritmo semplice basato sul metodo degli scambi.
Si fanno "salire" gli elementi più piccoli verso l'inizio del vettore scambiandoli con quelli adiacenti.

Si procede confrontando gli elementi a coppie e ogni qualvolta si trova una coppia non ordinata si scambiano di posto i due elementi.

Dato il vettore A[n], si opera confrontando A[1] con A[2] e se A[1] è maggiore di A[2], si effettua uno scambio tra i due; vengono poi confrontati A[2] con A[3], A[3] con A[4], …A[n-1] con A[n] scambiando di posto quegli elementi che non formano coppie ordinate.

Esempio: Sia A il vettore da ordinare

La condizione è falsa

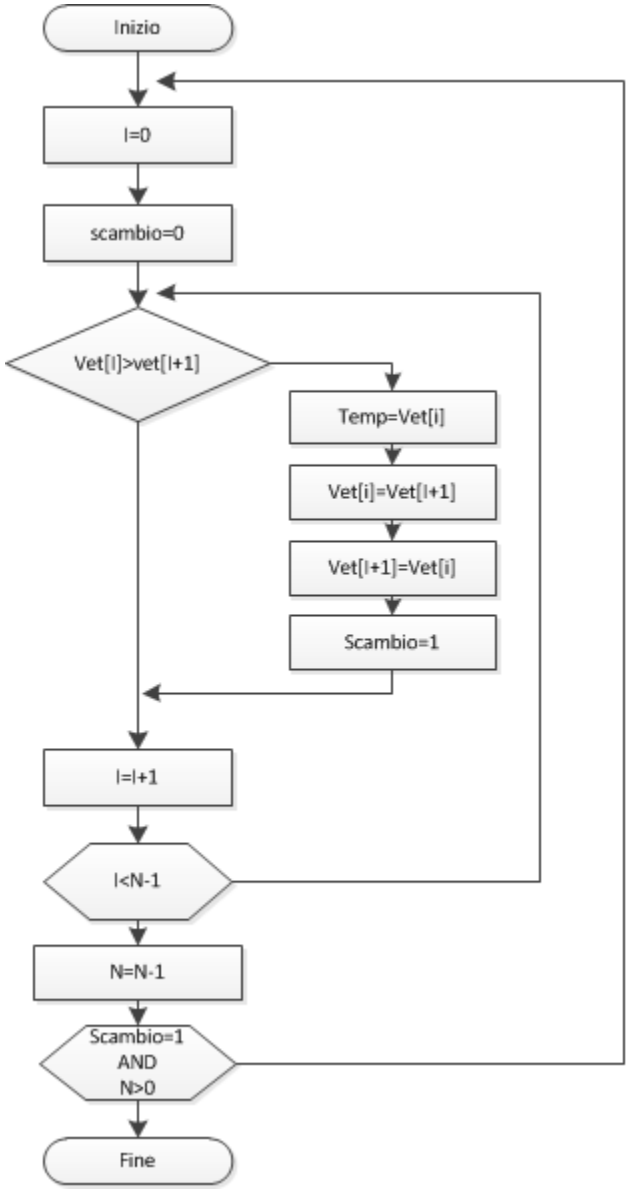
Scambio=0

Vettore Ordinato!

```
BubbleSort(A,n,scambio,temp)
 i:=1
  repeat
    scambio:= 0
      for j=i+1 to n do
         if (A[j] < A[j-1]) then
            temp:=A[j]
            A[j]:=A[j-1]
            A[j-1]:=temp
            scambio:=1
          endif
       endfor
  until scambio=1
end
```

Flowchart e Codifica in C

```
void Ordina(void){
Int I, Temp;
Char scambio
Do{
I=0;
Scambio=0;
  Do{
     If(Vet[I]>Vet[I+1])
      {
        Temp=Vet[I];
        Vet[I]=Vet[I+1];
        Vet[I+1]=Vet[I];
        scambio=1;
       }
     I=I+1;
   }while(i<N-1);
 N=N-1;
}while(scambio==1&&N>0);
}
```



Ordinamento per inserzione

- Un esempio di ordinamento per inserzione si applica quando si gioca a carte.
- Per ordinare le carte, in ordine crescente o decrescente, si estrae una carta, scalando quelle rimanenti, ed inserendo la carta estratta nel posto corretto.
- Il procedimento si ripete finché tutte le carte sono nella sequenza corretta.
- Il metodo di ordinamento ad inserzione trae lo spunto dall'idea che un vettore ordinato si ottiene inserendo le sue componenti una per una "al posto giusto".

Esempio Dato l' array :

11	7	3	8	6	5	9

per **i = 2** si ha :

11	**7**	3	8	6	5	9

si estrae l'elemento nella seconda posizione :

11		3	8	6	5	9

si sposta l'elemento precedente perchè, in questo esempio, risulta maggiore :

	11	3	8	6	5	9

si posiziona l'elemento estratto nella posizione corretta :

7	11	3	8	6	5	9

per **i = 3** si ha :

si estrae l'elemento nella terza posizione :

si spostano gli elementi precedenti che risultano maggiori :

si posiziona l'elemento estratto nella posizione corretta :

per **i = 4** si ha :

si estrae l'elemento nella quarta posizione :

si spostano gli elementi precedenti che risultano maggiori :

si posiziona l'elemento estratto nella posizione corretta :

per **i = 5** si ha :

si estrae l'elemento nella quinta posizione :

si spostano gli elementi precedenti che risultano maggiori :

si posiziona l'elemento estratto nella posizione corretta :

per **i = 6** si ha :

si estrae l'elemento nella sesta posizione :

si spostano gli elementi precedenti che risultano maggiori :

si posiziona l'elemento estratto nella posizione corretta :

per **i = 7** si ha :

si estrae l'elemento nella settima posizione :

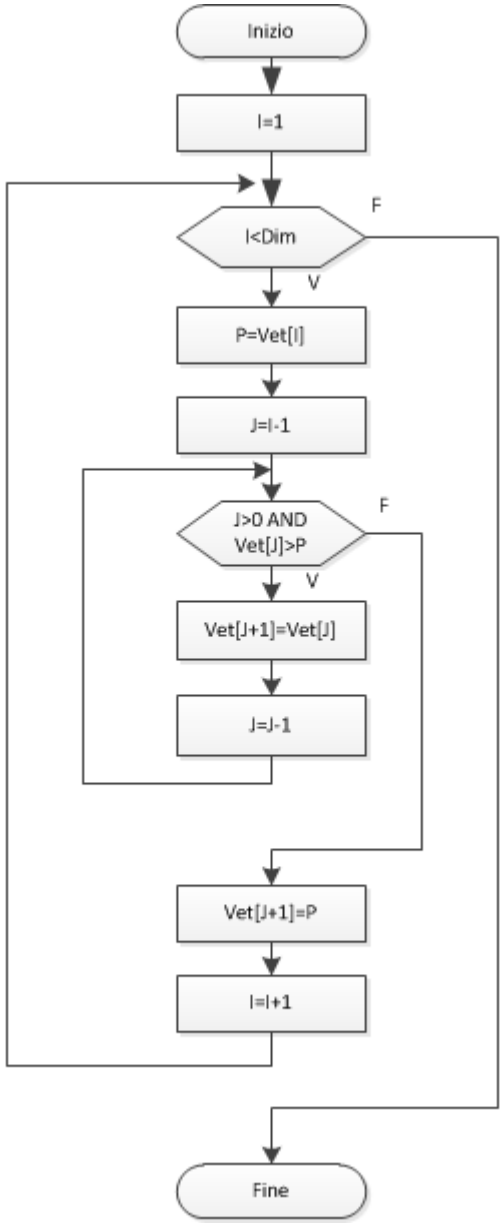
si spostano gli elementi precedenti che risultano maggiori :

si posiziona l'elemento estratto nella posizione corretta :

A questo punto il vettore risulta ordinato.

Flow Chart e codifica in C

```
void Ordina(void){
Int I,J,P;
I=1
While(I<Dim)
{
  P=Vet[i];
  J=I-1;
  while(J>0&&Vet[J]>P)
     {
     Vet[J+1]=Vet[J];
      J=J-1;
      }
 Vet[J+1=P;
 I=I+1
 }
}
```



Ordinamento per Selezione

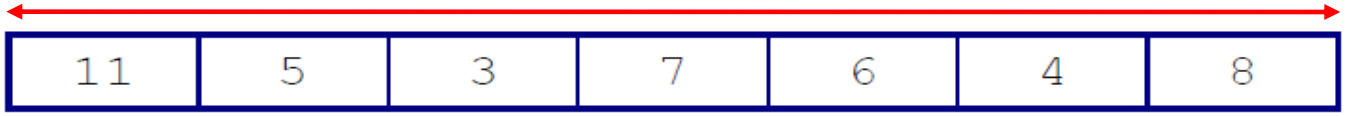
I passi da seguire sono i seguenti :

1) Posizionamento sul primo elemento dell'array
2) Ricerca dell'elemento più piccolo e scambio con il primo elemento dell'array
3) Posizionamento sul secondo elemento dell'array
4) Ricerca dell'elemento più piccolo tra gli N-1 elementi rimasti e scambio con il secondo elemento dell'array
5) Posizionamento sul terzo elemento dell'array
6) Ricerca dell'elemento più piccolo tra gli N-2 elementi rimasti e scambio con il terzo elemento dell'array
7) Tale procedimento viene ripetuto N-1 volte

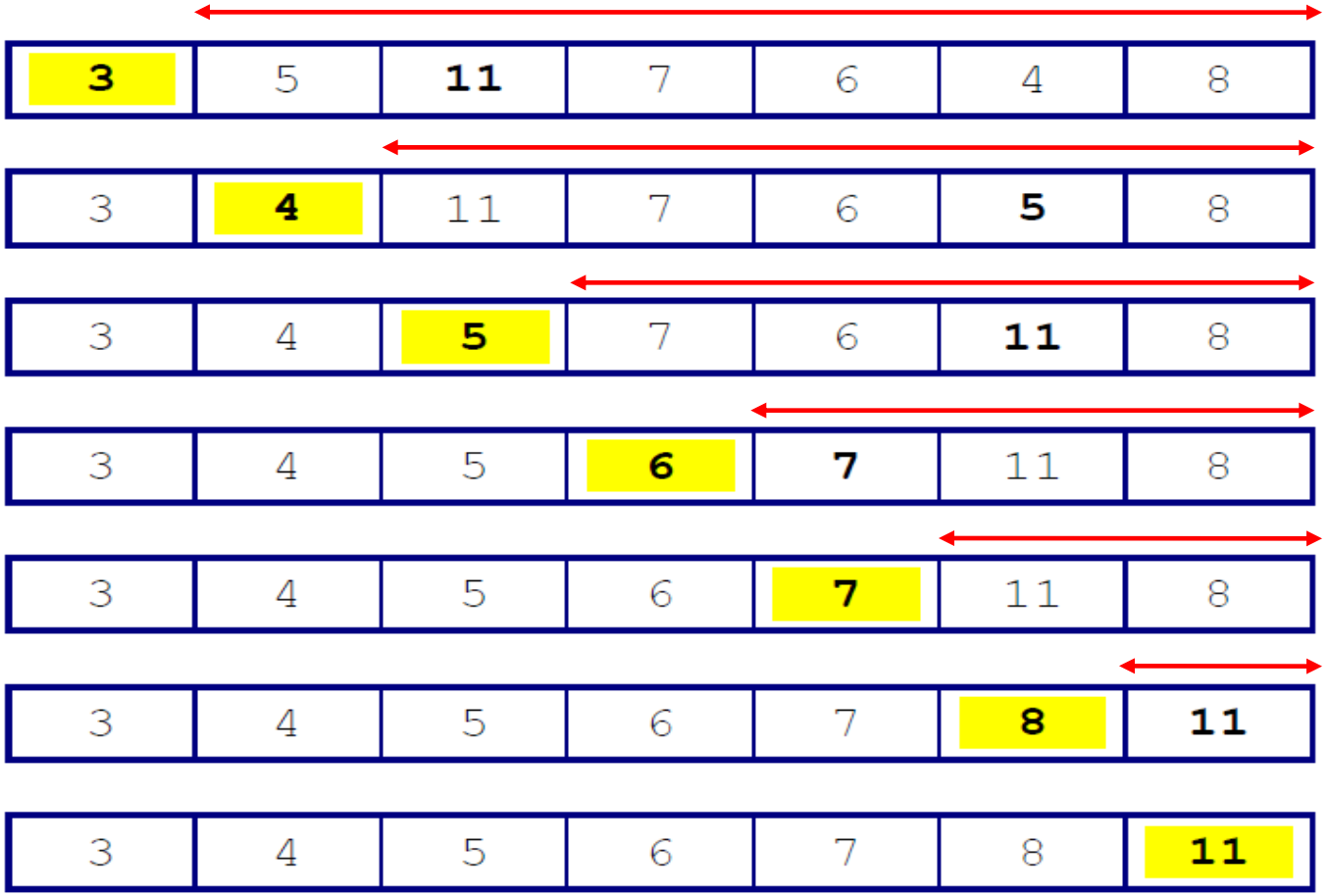
Osservazione : Per implementare l'Algoritmo abbiamo bisogno di 2 indici :

- Uno che tiene conto della posizione in cui si trova l'elemento da ordinare (primo, secondo, terzo, …)
- Uno che permette di scorrere l'array alla ricerca del valore maggiore

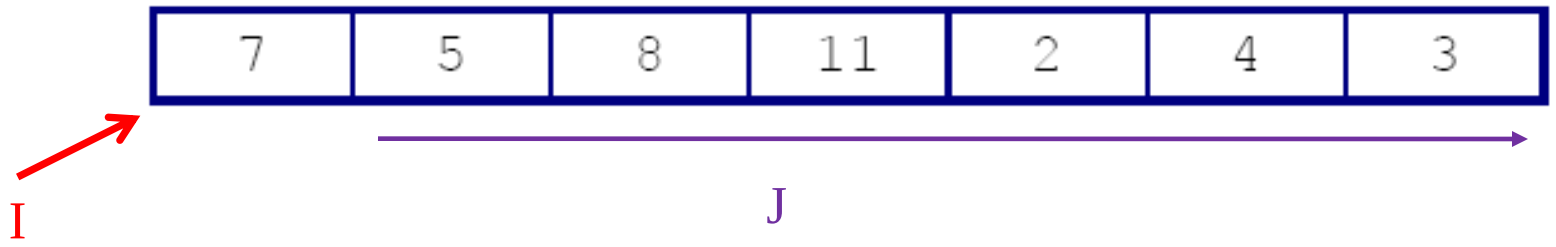
Esempio : dato l' array



i passi, per un ordinamento crescente, sono i seguenti :



- Per implementare l'Algoritmo si devono usare 2 indici :
  - Uno (I) che tiene conto della posizione in cui si trova l'elemento da ordinare (primo, secondo, terzo, …)
  - Uno (J) che permette di scorrere l'array alla ricerca del valore minore



Flow Chart e Codifica in C

```
void Ordina(void){
Int I,J,Pmin,Min;
I=0
While(I<Dim-1)
{
 Min=Vet[I];
 Pmin=I;
 J=I+1;
 while(J<Dim)
    {
    if(Vet[j]<Min){
      Min=Vet[J];
      Pmin=J;
       }
     J=J+1;
     }
 if(Pmin>I){
   Vet[Pmin]=Vet[I];
    Vet[I]=Min;
   }
 I=I+1;
}
}
```