



## Selection Sort (detto anche ordinamento per minimi successivi)

Supponiamo di avere il seguente vettore chiamato A con 10 elementi non ordinati.

A	15	19	91	12	28	82	22	8	19	24
	0	1	2	3	4	5	6	7	8	9

N=10

La strategia usata da questo algoritmo è la seguente:

E' necessario effettuare più scansioni del vettore.

- Alla prima scansione, si cerca il minimo del vettore e lo si scambia con l'elemento in posizione 0. per ricercare il minimo, assumo che esso sia in posizione 0 (metto 0 nella variabile posmin), poi se durante la scansione incontro un numero più piccolo, memorizzo nella variabile posmin la posizione del nuovo minimo. Quando la scansione sarà terminata scambierò l'elemento in posizione 0 con l'elemento in posizione posmin.
- Alla seconda scansione, si cerca il minimo della porzione di vettore da posizione 1 a posizione N-1 e lo si scambia con l'elemento in posizione 1. per ricercare il minimo, assumo che esso sia in posizione 1 (metto 1 nella variabile posmin), poi se durante la scansione incontro un numero più piccolo, memorizzo nella variabile posmin la posizione del nuovo minimo. Quando la scansione sarà terminata scambierò l'elemento in posizione 1 con l'elemento in posizione posmin.
- Alla terza scansione, si cerca il minimo della porzione di vettore da posizione 2 a posizione N-1 e lo si scambia con l'elemento in posizione 2... e così via di seguito

i indica la cella destinata ad accogliere il minimo

j è l'indice che scorre per cercare il minimo

posmin (inizialmente uguale a i) indica la posizione del minimo e sarà aggiornato ogni volta che troveremo che il valore puntato da j è minore del valore puntato da posmin

A	15	19	91	12	28	82	22	8	19	24	valori
	0	1	2	3	4	5	6	7	8	9	indici
	i	j									
	posmin										

vettore da ordinare in maniera crescente

N = 10

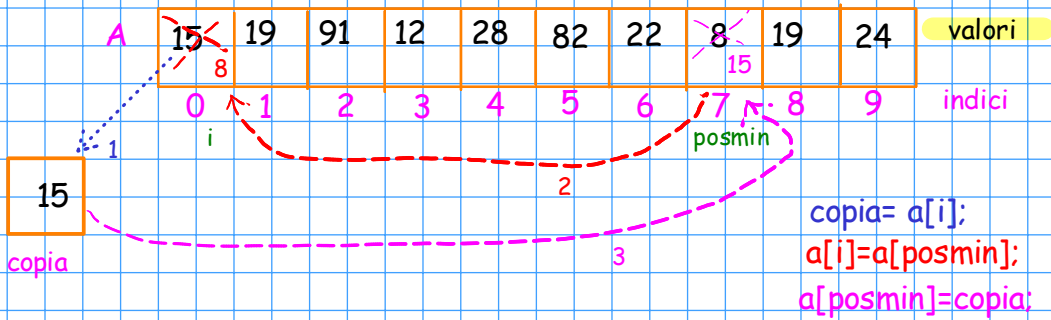
Facciamo scorrere l'indice j verso destra e ci fermiamo quando abbiamo raggiunto un valore più piccolo del valore puntato dall'indice posmin. A questo punto pongo posmin=j

A	15	19	91	12	28	82	22	8	19	24	valori
	0	1	2	3	4	5	6	7	8	9	indici
	i			j							
	posmin			posmin							
											j=10

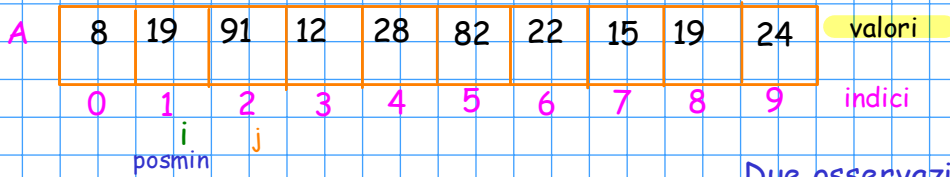


Alla fine della prima scansione la situazione è la seguente...

...e si esegue lo scambio tra la cella 0 e la cella posmin



configurazione del vettore dopo il primo scambio



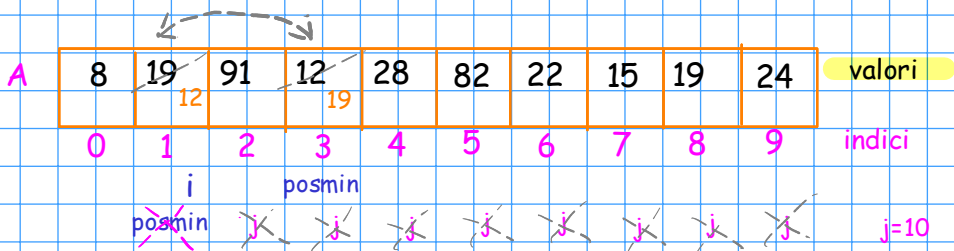
Due osservazioni:

- ✓ Il minimo del vettore si trova in posizione 0
- ✓ Abbiamo effettuato in tutto 9 (cioè N-1) confronti

Procediamo con la seconda scansione:

$i = 1$   
 $\text{posmin} = i$   
 $j = i + 1$  ...e si riprende a fare i confronti, come già descritto in precedenza

Alla fine della seconda scansione e prima del secondo scambio, lo stato delle variabili sarà il seguente:



procediamo con lo scambio

Abbiamo fatto 8 (cioè N-2) confronti

*next page*



Procedendo analogamente, alla fine il vettore sarà ordinato

A

8	12	15	19	19	22	24	28	82	91
0	1	2	3	4	5	6	7	8	9

Notare che l'algoritmo ha funzionato anche in presenza di elementi ripetuti

Quanti confronti saranno stati effettuati?

$$(N-1) + (N-2) + \dots + 2 + 1$$

la formula generale per calcolare il numero di confronti:

$$\frac{N(N-1)}{2} \quad \text{Nel nostro caso } \frac{10 * 9}{2} = 45 \text{ confronti}$$

Il caso peggiore si verifica quando i dati sono ordinati al contrario (cioè quando sono ordinati in maniera decrescente).

Il caso più favorevole si verifica quando i dati sono ordinati già in modo crescente

*next page*



Passiamo ora alla pseudocodifica

A	15	19	91	12	28	82	22	8	19	24
	0	1	2	3	4	5	6	7	8	9
	i	j								

i parte da 0 e finisce a  $N - 2$

All'inizio di una nuova scansione

$posmin=i$

j parte da  $i + 1$  e finisce a  $N - 1$

Per  $i=0$  fino a  $i=N-2$  compreso

$posmin=i$

Per  $j=i+1$  fino a  $j=N-1$  compreso

Se  $a[posmin] > a[j]$  allora

$posmin=j$

fine se

fine per

se  $i \neq posmin$  scambia  $a[i]$  con  $a[posmin]$

fine per

scambia(a,i,j)

```
copia = a[i]
a[i] = a[j]
a[j] = copia
```

next page



Di seguito è riportato il codice completo in linguaggio C

```

/*****
selection sort
*****/
#include <stdio.h>

#define MAX 100
int nconfronti=0, nscambi=0;
int carica_vettore(int a[]) {
    int n, i;
    printf("Quanti elementi?: ");
    scanf("%d", &n);

    for (i=0; i<n; i++) {
        printf("elemento %d: ", i);
        scanf("%d", &a[i]);
    }
    return(n);
}

void stampa_vettore(int a[],int p, int u) {
    int i;
    for (i=p; i<=u; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");
    return;
}

void scambia(int a[],int x,int y){
    int copia;
    copia=a[x];
    a[x]=a[y];
    a[y]=copia;
}

void SelectionSort(int a[], int n){
    int i,j, posmin;

    for(i=0;i<=n-2;i++){
        posmin=i;
        for (j=i+1; j<=n-1;j++){
            nconfronti++;
            if(a[posmin]>a[j]){
                posmin=j;
            }
        }
        if(i!=posmin){
            nscambi++;
            scambia(a,i,posmin);
        }
    }
}
}

```



```
iint main(){  
  
    //int n, a[MAX],i;  
    //n = carica_vettore(a);  
    int n, i;  
    int a[]={52,25,35,53,63,36,66,48,12,12,88,84,48,78,26};  
    n = 15;  
  
    printf("Vettore iniziale:\n");  
    stampa_vettore(a,0,n-1);  
    SelectionSort(a, n);  
    printf("Vettore ordinato con selection sort:\n");  
    stampa_vettore(a,0,n-1);  
    printf("sono stati eseguiti %d confronti\n",nconfronti);  
    printf("sono stati eseguiti %d scambi\n",nscambi);  
    return 0;  
}
```