

مثال : عبارت زیر را بصورت prefix و postfix بنویسید.

$$\sqrt{a^2 - bc} \Rightarrow (a \uparrow 2 - b \times c) \uparrow (1/2)$$

$$\text{postfix} = a2 \uparrow bc \times -12 / \uparrow$$

$$\text{prefix} = \uparrow - \uparrow a2 \times bc / 12$$

### مرتب‌سازی

در مرتب‌سازی تعدادی عنصر که از ورودی داده شده‌اند را بر اساس کلیدشان بصورت صعودی یا نزولی مرتب می‌کنیم.

### مرتب‌سازی انتخابی (Selection Sort)

در مرتب‌سازی انتخابی یک آرایه  $n$  عنصری  $(A[1..n])$ ،  $n - 1$  بار پیمایش می‌شود. در هر پیمایش بزرگترین عنصر در محل درست خود یعنی انتهای آرایه قرار می‌گیرد. با این روش آرایه از انتها مرتب می‌شود. در مرتب‌سازی انتخابی می‌توان با انتخاب کوچکترین عنصر در هر پیمایش و قرار دادن آن در محل درست خود یعنی ابتدای آرایه در هر پیمایش، مرتب‌سازی را از ابتدای لیست انجام داد.

مثال :

10	5	8	20	25	12	
10	5	8	20	12	25	پویش اول
10	5	8	12	20	25	پویش دوم
10	5	8	12	20	25	پویش سوم
8	5	10	12	20	25	پویش چهارم
5	8	10	12	20	25	پویش پنجم

برنامه کلی مرتب‌سازی انتخابی به شرح ذیل می‌باشد :

```

for (i = n ; i > 1 ; -- 1)
{
    max = A[1] ;
    index = 1 ;
    for (i = 2 ; j <= i ; ++ j)
    if (A[j] > max)
    {
        max = A[j] ;
        index = j ;
    }
    A[index] = A[i] ;
    A[i] = max ;
}

```

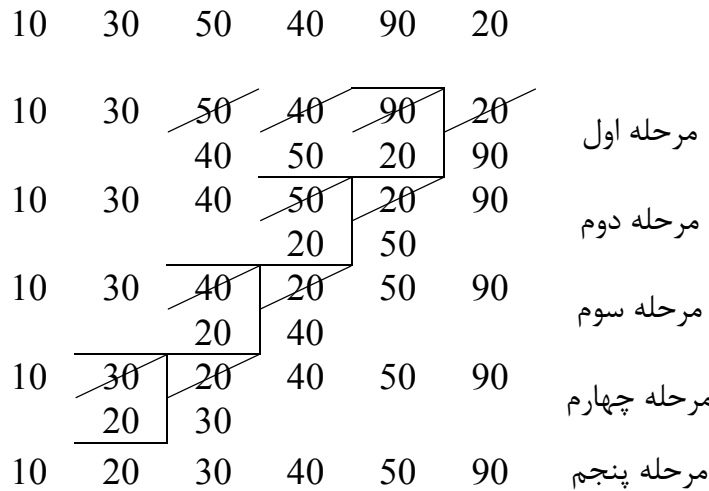
مثال :

3	2	5	1	
1	2	3	4	
i	j	n	max	index
4	2	4	3	1
3	4		5	3
	4		3	1
	2			
	3			

نکته : در مرتب‌سازی انتخابی ، حداکثر و حداقل  $n^2$  مقایسه داریم. حداقل جابجایی صفر و حداکثر جابجایی نیز  $n$  بار خواهد بود.

**مرتب‌سازی حبابی (Bubble Sort)**

در مرتب‌سازی حبابی یک آرایه  $n$  عنصری  $(A[1..n])$ ،  $n - 1$  بار پیمایش می‌شود و در هر پیمایش دو عنصر متوالی با یکدیگر مقایسه شده که در صورت لزوم جابجا خواهند شد. در هر پیمایش، طول آرایه پیمایش شده نسبت به مرحله قبل یکی کم می‌شود.



مرتب‌سازی از انتهای لیست

```
for (i = 1 ; i < n ; ++ i)
    for (j = 1 ; j <= n ; ++ j)
        if (A[j] > A[j + 1])
            swap (A[j] , A[j + 1]) ;
```

مرتب‌سازی از ابتدای لیست

```
for (i = 1 ; i < n ; ++ i)
    for (j = n ; j >= i ; -- j)
        if (A[j] < A[j - 1])
            swap (A[j] , A[j - 1]) ;
```

مثال :

5	3	7	2
---	---	---	---

1 2 3 4

2	5	3	7
---	---	---	---

پویش اول

2	3	5	7
---	---	---	---

پویش دوم

i	j	n
1	4	4
2	3	
3	2	
	1	
	4	

« الگوریتم متعادل است »