ساختمان داده ها و الگوریتم ها

پیچیدگی زمانی و مکانی

محمدحسین اولیائی

مجتمع آموزش عالی گناباد

روش های تحلیل الگوریتم ها

هدف از تحلیل الگوریتم‌ها:

- **بررسی رفتار الگوریتم قبل از پیاده‌سازی، از نظر زمان اجرا و مقدار حافظه‌ی مصرفی**
- **مقایسه‌ی الگوریتم‌ها با هم از نظر کارایی**

زمان اجرا

عوامل زیر در زمان اجرای یک برنامه موثرند:

۱) سرعت سخت افزار

۲) نوع کامپایلر

۳) اندازه‌ی داده‌ی ورودی مسئله

۴) ترکیب داده‌ی ورودی

۵) پیچیدگی الگوریتم

۶) پارامترهای دیگر که تاثیر ثابت در زمان اجرا دارند

زمان اجرا

زمان اجرای الگوریتم: $T(n)$، n **اندازه‌ی ورودی مسئله**

توجه:

- **ممکن است چند داده‌ی ورودی داشته باشیم: مثلاً**
- **یک گراف، تعداد راس‌ها برابر** n، **تعداد یال‌ها برابر** m، **زمان اجرا** $T(n, m)$
- **چند پارامتر، انتزاع** (abstraction)

چند مثال

```
float power(int a,int n)
{ float temp=1;
 for (int i=0; i<n; i++)
     temp*=a;
return temp;
}
```

چند مثال

```
float power(int a,int n)
{ float temp=1;
 for (int i=0; i<n; i++)
     temp*=a;
return temp;
}
```

$$\boldsymbol{T(n) = 2n + 3}$$

چند مثال

```
void Fill_Array(int a[],int L)
{ int m;
for (int i=0; i<L; i++){
  cin>>m;
  a[i]=m;}
}
```

چند مثال

```
void Fill_Array(int a[],int L)
{ int m;
for (int i=0; i<L; i++)
  cin>>m;
  a[i]=m;
}
```

چند مثال

```
int factorial (int n)
{ int m=1;
for (int i=1; i<=n; i++)
  m*=i;
return m;
}
```

چند مثال

```
void ReverseArray (int a[], int n)
{ int temp;
for (int i=0; i<n/2; i++)
   {
      temp=a[i];
      a[i]=a[n-1-i];
      a[n-1-i]=temp;
   }
}
```

چند مثال

```
Void FillMatrix (int a[][], int n, int m)
{
 for (int i=0; i<m; i++)
    for (int j=0; j<n; j++)
     cin>>a[i][j];
}
```

چند مثال

```
Void FillMatrix (int a[][], int n, int m)
{
 for (int i=0; i<m; i++)
    for (int j=0; j<n; j++)
     cin>>a[i][j];
}
```

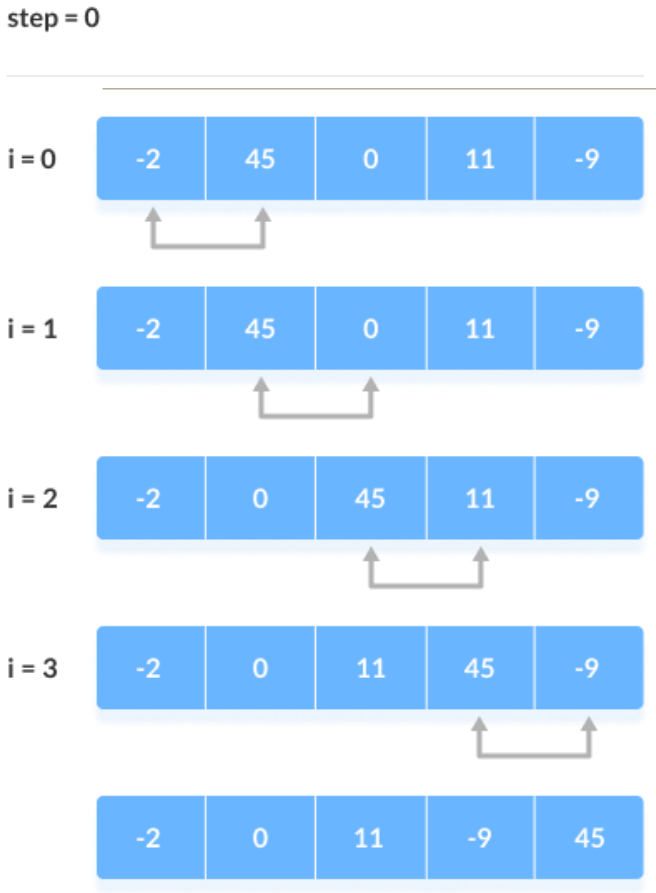
$m+1$

$m(n+1)$

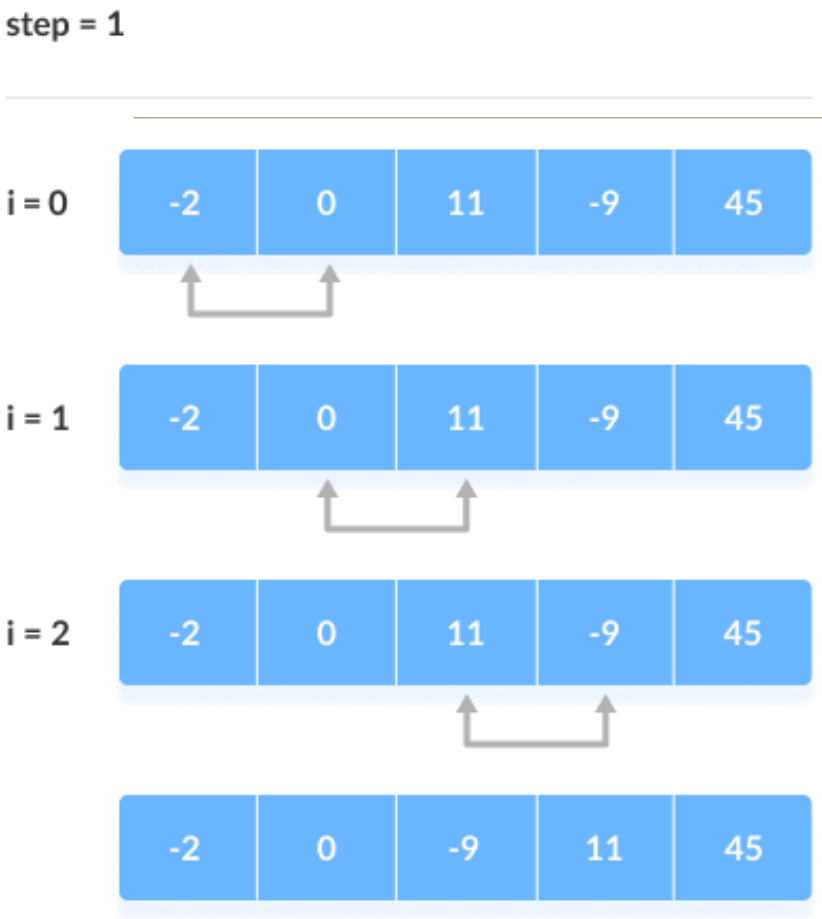
$m(n)$

$2mn+2m+1$

مرتب سازی حبابی

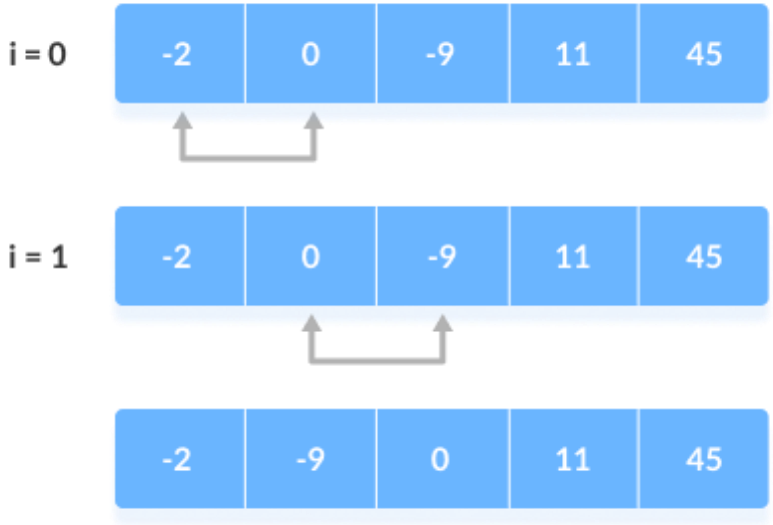


مرتب سازی حبابی



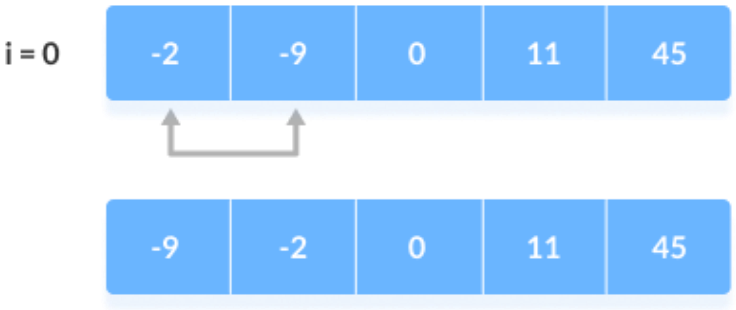
مرتب سازی حبابی

step = 2



مرتب سازی حبابی

step = 3



مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

Cycle	Number of Comparisons
1st	(n)
2nd	(n-1)
3rd	(n-2)
.....	.....
last	2

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

Cycle	Number of Comparisons
1st	(n)
2nd	(n-1)
3rd	(n-2)
.....	.....
last	2

نکته:

$$1 + 2 + 3 + \cdots n = n(n+1)/2$$

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

Cycle	Number of Comparisons
1st	(n-1)
2nd	(n-2)
3rd	(n-3)
.....	.....
last	1

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

Cycle	Number of Comparisons
1st	(n-1)
2nd	(n-2)
3rd	(n-3)
.....	.....
last	1

(n - 1) + (n - 2) + (n - 3) +.....+ 1 = n(n - 1) / 2

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
      for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
          swap(&arr[j], &arr[j+1]);
}
```

$n - (n - 2) - 1 = 1$

Cycle	Number of Comparisons
1st	(n-1)
2nd	(n-2)
3rd	(n-3)
…….	……
last	1

(n - 1) + (n - 2) + (n - 3) +.....+ 1 = n(n - 1) / 2

$$n + \frac{n(n+1)}{2} - 1 + \frac{n(n-1)}{2} + \frac{n(n-1)}{2} = n + \frac{n(n+1)}{2} - 1 + \mathrm{n(n-1)}$$

یک حالت خاص

```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

فرض کنیم آرایه ورودی از ابتدا مرتب باشد:

3	6	12	32	33	45	700	902

مرتب سازی حبابی

```
void bubbleSort(int arr[], int n) {
  for (int i= 0; i< n - 1; ++i) {
    int swapped = 0;
    for (int j = 0; j < n - i- 1; ++j) {
      if (arr[i] > arr[i + 1]) {
          swap(&arr[j], &arr[j+1]);
                      swapped = 1;}
      }
     if (swapped==0)
         return;
   }
}
```

$$1 + 2 + 3 + \cdots n = n(n+1)/2$$

$$2 + 3 + \cdots n = \frac{n(n+1)}{2} - 1$$

نکته

چند رابطه کاربردی:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}.$$