# روش های تحلیل الگوریتم

# زمان اجرای الگوریتم ها

الگوریتم عبارت است از:

مجموعه ای از دستورات و دستورالعمل ها برای حل مسئله،که شرایط زیر رو باید داشته باشد:

1.دقیق باشد
2.مراحل آن به ترتیب انجام پذیرد
3.پایان پذیرباشد

# زمان اجرای الگوریتم ها

عوامل دخیل در زمان اجرای برنامه عبارتند از:

- سرعت سخت افزار
- نوع کامپایلر
- اندازه داده ورودی
- ترکیب داده های ورودی
- پیچیدگی زمانی الگوریتم
- پارامترهای دیگر که تاثیر ثابت در زمان اجرا دارند.

# پیچیدگی زمانی الگوریتم

**تابع T(n)**

**مثال :**

**تعداد راس ها n**

**تعداد یال ها m**

**T(n,m)**

**برای محاسبه تابع T(n) برای یک الگوریتم موارد زیر را باید در محاسبات در نظر بگیریم:**

- **زمان مربوط به اعمال جایگزینی که مقدار ثابت می باشند**
- **زمان مربوط به انجام اعمال محاسبات که مقدار ثابتی دارند.**
- **زمان مربوط به تکرار تعدادی دستور یا دستورالعمل(حلقه ها)**
- **زمان مربوط به توابع بازگشتی**

# الگوریتم مرتب سازیInsertion

## «مرتب سازی درجی»

**مساله:تابع الگوریتم مرتب سازی Insertion :**

**ورودی:A یک آرایه**

**n طول آرایه یا تعداد عناصر آرایه**

**خروجی: لیست مرتب از داده ها(A,n)**

```
Void Insertion_Sort (element type A[],int n)
{
    element type   key;
(1)  for(int k=1;k<=n ; k++){
(2)  key=A[k];
(3)  i=k-1;
(4)  while(i>=0)&&(A[i]>key)){
(5)          A[i+1]=A[i];
(6)          i=i-1;
       }
(7)         A[i+1]=key;
         }
   }
```

## مثال

1) 12,11,13,5,6

1) 11,12,13,5,6

3) 5,11,12,13,6

4) 5,6,11,12,13

# الگوریتم مرتب سازیInsertion

| سطر | هزینه | تعداد |
|---|---|---|
| 1 | $C_1$ | $n$ |
| 2 | $C_2$ | $n-1$ |
| 3 | $C_3$ | $n-1$ |
| 4 | $C_4$ | $\sum_{k=1}^{n-1} t_k$ |
| 5 | $C_5$ | $\sum_{k=1}^{n-1} (t_k - 1)$ |
| 6 | $C_6$ | $\sum_{k=1}^{n-1} (t_k - 1)$ |
| 7 | $C_7$ | $n-1$ |

$$T(n) = C_1 n + (C_2 + C_3 + C_7)(n-1) + C_4 + \sum_{k=1}^{n-1} t_k + (C_5 + C_6)\sum_{k=1}^{n-1} (t_k - 1)$$

$$T(n) = An + B\sum_{k=1}^{n-1} t_k + c$$

$$\begin{aligned} T(n) &= An + B\sum_{k=1}^{n-1} t_k + c \\ &= An + B\left(\frac{n(n+1)}{2} - 1\right) + c \\ &= an^2 + bn + c \end{aligned}$$

# بررسی حالت های مختلف یک الگوریتم

برای محاسبه زمان اجرای یک الگوریتم حالات زیر را در نظر می گیرند:

- بررسی بدترین حالت(worst Case)
- بررسی حالات متوسط(average Case)
- بررسی بهترین حالت(best Case)

1,2,3,4

4,3,2,1

بررسی حالت های مختلف زمان اجرای الگوریتمInsertion_Sort:

$$\bar{T}(n) = An + C + \frac{1}{n!}\sum_{i=0}^{n-1}\sum_{k=1}^{n-1} Bt_{k,i}$$

$$\bar{T}(n) = An + C + B\sum_{k=1}^{n-1}\bar{t}_k$$

$$\bar{t}_k = \sum_{i=0}^{k-1}\left(\frac{1}{k}(K-i-1)\right) = \frac{1}{k}\times\frac{k(k+1)}{2} = \frac{k+1}{2}$$

$$\bar{T}(n) = An + C + B\sum_{k=1}^{n-1}\frac{k+1}{2} = an^2 + bn + c$$

# مرتبه اجرایی الگوریتم

قطعه برنامه زیر را در نظر بگیرد:

```
(1) x=0;
(2) For (i=0; i<n;i++)
(3)     x++;
```

تابع زمانی قطعه کد بالا:

| سطر | زمان | تعداد |
|---|---|---|
| 1 | $C_1$ | ۱ |
| 2 | $C_2$ | n+۱ |
| 3 | $C_3$ | n |

با توجه به جدول، $T(n)$ برابر است با:

$$T(n) = C_1 + C_2(n+1) + C_3 n$$

حالا c را بیشترین مقدار c1,c2,c3 در نظر میگیرم بنابراین:

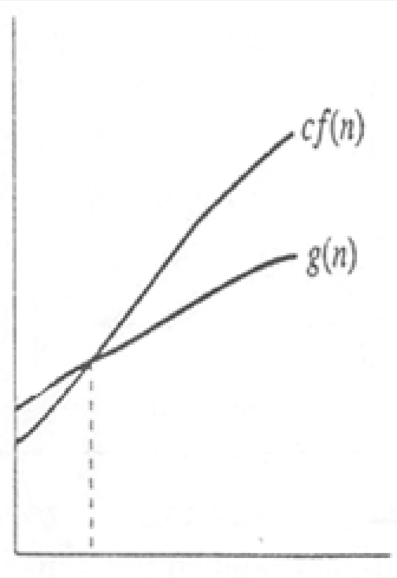
$$T(n) = C(2n+2)$$

# Big-oh(اوی بزرگ)

عبارت $g(n) \in O\big(f(n)\big)$

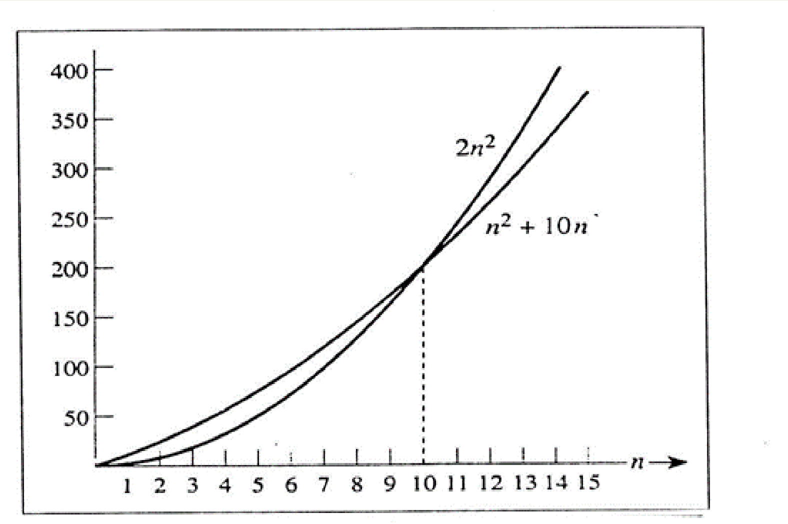
یعنی:برای تابع پیچیدگی مفروض f(n)، O(f(n)) به مجموعه ای از توابع اشاره دارد که برای آنها

ثابتهای c و $n_0$ وجود دارند،بطوریکه برای همه $n \geq n_0$ داریم $g(n) \leq cf(n)$




## مثال

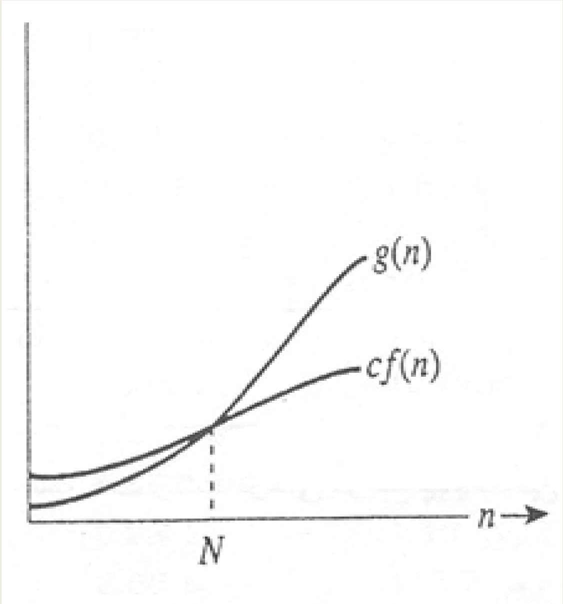
$n^2 + 10n \in O(n^2)$



# امگا بزرگBig-Omega

عبارت $g(n) \in \Omega(f(n))$

یعنی:برای تابع پیچیدگی مفروض f(n)، Ω(f(n)) به مجموعه ای از توابع اشاره دارد که برای آنها ثابتهای c و $n_0$ وجود دارند،بطوریکه برای همه $n \ge n_0$ داریم $g(n) \ge cf(n)$

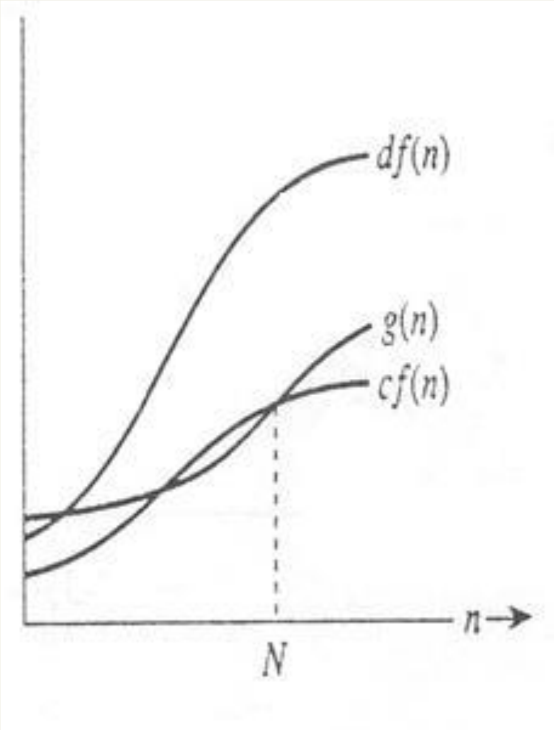



g(n) ∈ Ω(f(n))

# تتا

عبارت $g(n) \in \theta(f(n))$ یا $g(n) = \theta(f(n))$

یعنی: $g(n) \in O(f(n))$ $g(n) \in \Omega(f(n))$




$g(n) \in \theta(f(n))$