



ساختمان داده ها و الگوریتم ها

پیچیدگی زمانی (حل مثال)

محمد حسین اولیائی

پیچیدگی زمانی قطعه کد زیر:

```
sum=0;
for(i=1;i<=n;i++)
  for(j=1; j<=m;j++)
    sum++;
```

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1;i<=1000;i++)  
  //instructions
```

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1;i<=n;i++)  
  for(j=1; j<=i;j++)  
    //instructions
```

پیچیدگی زمانی قطعه کد زیر:

```
sum=0;
for(i=0;i<n;i++)
  for(j=0; j<i;j++)
    for(k=0;k<n;k++)
      sum++;
```

پیچیدگی زمانی قطعه کد زیر:

```
sum=0;
for(i=0;i<n;i++)
  for(j=0; j<i;j++)
    for(k=0;k<n;k++)
      sum++;
```

```
sum=0;
for(i=0;i<n;i++)
  for(j=0; j<i;j++)
    for(k=0;k<30000;k++)
      sum++;
```

پیچیدگی زمانی قطعه کد زیر:

```
for i=1 to n
  for j=1 to i
    for k=1 to n^2
      sum++
```

کدام یک از عبارات زیر غلط می باشد:

$$10^n + n^{20} \in \theta(n^n)$$

$$(\log n)! \in O(n!)$$

$$10n^3 + 16n^2 + 7n \in \Omega(\log n)$$

کدام یک از عبارات زیر غلط می باشد:

$$n! \in O(n^n)$$

$$(n + 1)(n^2 - 2n + 1) \in O(n^2 \log n)$$

$$\sqrt{n} \in O(\log n)$$

$$n^3 \log n = O(n^{3+\varepsilon})$$

پیچیدگی زمانی قطعه کد زیر:

```
while n>0  
    n=n/10
```

پیچیدگی زمانی قطعه کد زیر:

For(i=1;i<=n;i+=2);

پیچیدگی زمانی قطعه کد زیر:

For($i=1; i \leq n; i*=2$);

پیچیدگی زمانی قطعه کد زیر:

```
for i=0 to n
{
    j=n
    while j>0
        j=j/2
}
```

توابع زیر را بر حسب افزایش مرتبه از چپ به راست مرتب کنید

$$n^{1000}, (1.005)^n, n!, n^\varepsilon$$

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1; i<=n; i++)  
{  
    for(j=1;j<=m;j++)  
        x++;  
n--;  
}
```

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1; i<=n; i++)      m
{                          m
  for(j=1; j<=m; j++)    m
    x++;
n--;                      m
}
```



$$m + m + m + \dots m \cong n/2 \times m$$

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1; i<=n; i++)  
{  
    for(j=1; j<=n; j++)  
        x++;  
n--;  
}
```

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1; i<=n; i++)  
{  
  for(j=1; j<=n; j++)  
    x++;  
  n--;  
}
```

n

$(n - 1)$

$(n - 2)$

$n/2$



$$n + (n - 1) + (n - 2) + \dots + n/2 \cong n/2 \times n$$

فرض کنید f و g دو تابع دلخواه بوده و داشته باشیم: $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$

کدامیک از موارد زیر صحیح است:

$$g(n) \in O(f(n)), f(n) \in O(g(n))$$

$$f(n) \in \Omega(g(n)), f(n) \notin \Theta(f(n))$$

$$f(n) \in \Theta(g(n)), g(n) \notin \Omega(f(n))$$

$$f(n) \in O(g(n)), g(n) \notin \Omega(f(n))$$

فرض کنید f و g دو تابع دلخواه بوده و داشته باشیم $f(n) \in O(g(n))$ کدام مورد می تواند صحیح باشد:

$$2^{g(n)} = O(2^{f(n)})$$

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

$$2^{g(n)} = \Omega(2^{f(n)})$$

$$\log(g(n)) = O(\log(f(n)))$$

پس از اجرای کد زیر، مقدار نهایی X چقدر است؟

```
x=0;
for i=1 to n
{
  j=1;
  while(j<n)
  {
    x++;
    j=j*2;
  }
}
```

پس از اجرای کد زیر، مقدار نهایی X چقدر است؟

```
x=0;
for i=1 to n
{
  for j=1 to n
    x++;
  j=1;
  while(j<n)
  {
    x++;
    j=j*2;
  }
}
```

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1; i<=n; i++)  
{  
    for(j=1; j<=n; j=j+i)  
        x++;  
}
```

پیچیدگی زمانی قطعه کد زیر:

```
for(i=1; i<=n; i++)  
{  
  for(j=1; j<=n; j=j+i)       $n + n/2 + n/3 + \dots + 1 \rightarrow n(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n})$   
    x++;  
}
```

$$\sum_{r=1}^n \frac{1}{r} \approx \log n$$

تعیین اشتراک برای دو لیست وقتی که:

هر دو مرتب باشند

هر دو نامرتب باشند

15	23	2	17	5
----	----	---	----	---

53	2	20	15	0	122	17	5	9
----	---	----	----	---	-----	----	---	---

تعیین اشتراک برای دو لیست وقتی که:

```
For(i=1;i<=n;i++)  
  for(j=1;j<=m;j++)  
    if (a[i]==b[j])  
      cout<<a[i];
```

هر دو مرتب باشند

هر دو نامرتب باشند

15	23	2	17	5
----	----	---	----	---

53	2	20	15	0	122	17	5	9
----	---	----	----	---	-----	----	---	---

تعیین اشتراک برای دو لیست وقتی که:

هر دو مرتب باشند

هر دو نامرتب باشند

2	15	20	170	502
---	----	----	-----	-----

5	7	34	87	100	170	187	502	900
---	---	----	----	-----	-----	-----	-----	-----

تعیین اشتراک برای دو لیست وقتی که:

هر دو مرتب باشند

هر دو نامرتب باشند

```
For(i=1,j=1;i<=n && j<=m;)
{
  if(a[i]==b[j])
  { cout<<a[i];
    i++;j++;
  }
  if(a[i]<b[j])i++;
  if(a[i]>b[j]) j++;
}
```

2	15	20	170	502
---	----	----	-----	-----

5	7	34	87	100	170	187	502	900
---	---	----	----	-----	-----	-----	-----	-----