ساختمان داده ها

مدرس:
فرشید شیرافکن

دانشجوی دکتری دانشگاه تهران
(کارشناسی و کارشناسی ارشد: کامپیوتر نرم افزار) (دکتری: بیو انفورماتیک)

سرفصل

1. مرتبه اجرایی (پیچیدگی اجرایی)
2. توابع بازگشتی
3. آرایه
4. صف و پشته
5. لیست پیوندی
6. درخت
7. گراف
8. مرتب سازی
9. درهم سازی

منابع

- کرمن
- هورویتز
- دکتر قدسی
- لیپ شوتز
- شیرافکن

پیچیدگی اجرایی

پیچیدگی یک الگوریتم، تابعی است که مدت زمان اجرای استفاده شده توسط الگوریتم را بر حسب تعداد داده‌های ورودی n اندازه می‌گیرد.

notation	name
O(1)	constant
O(n)	linear
O(logn)	logarithmic
O(n^2)	quadratic
O(n^c)	polynomial
O(c^n)	exponential
O(n!)	factorial

مرتبه اجرایی توابع چند جمله ای

در صورتی که داشته باشیم:

$$f(n) = n^m + n^{m-1} + \ldots + n^2 + n + c \Rightarrow f(n) = O(n^m)$$

مثال:

$$f(n) = 5n^2 - 3n + 4 \Rightarrow O(n^2)$$

$$O(1) < O(\lg n) < O(n) < O(n \lg n) < O(n^2) < O(2^n) < O(n!) < O(n^n)$$

$$\log_2^n = \lg n$$

نمادهای پیچیدگی اجرایی

اوی بزرگ

عبارت $f(n) \in O(g(n))$ یعنی: برای تابع پیچیدگی مفروض g(n) ، O(g(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: $f(n) \leq cg(n)$

امگا بزرگ

عبارت $f(n) \in \Omega(g(n))$ یعنی: برای تابع پیچیدگی مفروض g(n) ، $\Omega(g(n))$ 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: $f(n) \geq cg(n)$

تتا

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

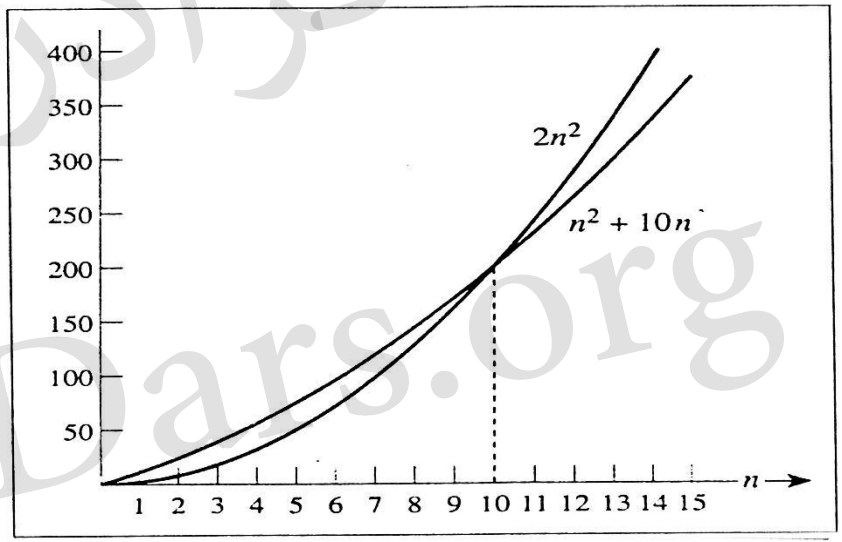
مثال

نشان دهید : $n^2 + 10n \in O(n^2)$

$n^2 + 10n \le cn^2$

$n^2 + 10n \le 2n^2$

$n^2 + 10n = 2n^2 \Rightarrow 10n = n^2 \Rightarrow n = 10$



مثال

نشان دهید : $\frac{n(n-1)}{2} \in O(n^2)$

حل:

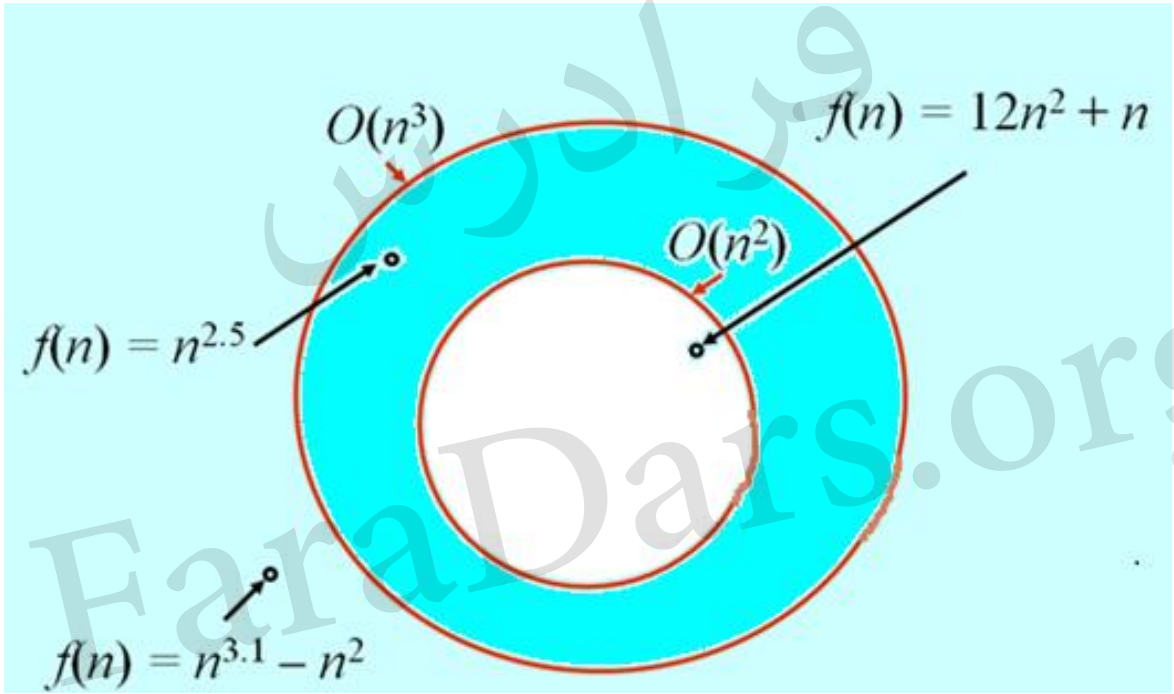
$$\frac{n(n-1)}{2} \le cn^2$$

$$\frac{n(n-1)}{2} \le \frac{n^2}{2}$$

$$\frac{n(n-1)}{2} = \frac{n^2}{2} \Rightarrow n = 0$$

یعنی : $n_0 = 0, c = \frac{1}{2}$

مثال



مثال

نشان دهید : $\frac{n(n-1)}{2} \in \Omega(n^2)$

حل:

$$\frac{n(n-1)}{2} \geq cn^2$$

$$\frac{n^2}{2} - \frac{n}{2} \geq \frac{1}{4}n^2$$

$$\frac{n^2}{2} - \frac{n}{2} = \frac{n^2}{4} \Rightarrow n = 2$$

یعنی: $n_0 = 2, c = \frac{1}{4}$

مثال

نشان دهید که : $\frac{n(n-1)}{2} \in \theta(n^2)$

جواب:

در مثال های قبل دیدیم که $\frac{n(n-1)}{2} \in O(n^2)$ و $\frac{n(n-1)}{2} \in \Omega(n^2)$ بنابراین :

$$\frac{n(n-1)}{2} \in \theta(n^2)$$

خواص توابع رشد

۱) **بازتابی** :

$$f(n) = O(f(n)), \qquad f(n) = \Omega(f(n)), \qquad f(n) = \theta(f(n))$$

۲) **تراگذاری** .

مثلا برای تتا :

$$\left.\begin{array}{l} f(n) = \theta(g(n)) \\ g(n) = \theta(h(n)) \end{array}\right\} \Rightarrow f(n) = \theta(h(n))$$

۳) تتا خاصیت **تقارن** دارد:

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

۴) O خاصیت تقارن ترانهاده دارند:

$$f(n) = O(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$$

پیچیدگی دستور if

به دستور if زیر توجه کنید:

```
if(cond)
    block1
else
    block2
```

زمان اجرا در بدترین حالت برابر است با:

Max(time(block1) , time(block2))

مرتبه اجرایی حلقه های ساده

حلقه for زیر را در نظر بگیرید: (a<=b)

```
for (i=a ; i<=b ; i=i+k )
    s;
```

```
for (i=b ; i>=a ; i=i-k )
    s;
```

تعداد تکرار : $\frac{b-a+1}{k}$

اگر این مقدار اعشاری بود، حد بالای آن را در نظر بگیرید.

منظور از s ، توالی از دستورها (sequence of statement) است.

مثال

تعداد تکرار دستور داخل حلقه را مشخص کنید.

```
for ( i=1 ; i<=n ; i=i+1 )
        s;
```

جواب:

تعداد تکرار : $\frac{n-1+1}{1} = n$

مرتبه اجرایی : o(n)

مثال

تعداد تکرار دستور داخل حلقه را مشخص کنید.

```
for ( i=3 ; i<=n ; i=i+2 )
        s;
```

جواب:

تعداد تکرار : $\frac{n-3+1}{2} = \frac{n}{2} - 1$

مرتبه اجرایی : o(n)

مثال

تعداد تکرار دستور داخل حلقه را مشخص کنید.

```
for ( i=9 ; i<3n+4 ; i=i+5 )
        s;
```

جواب:

تعداد تکرار: $\frac{3n+4-9}{5} = \frac{3n}{5} - 1$

مرتبه اجرایی : o(n)

مرتبه ثابت

مرتبه اجرایی دستور x=x+1 را مشخص کنید.

```
i=n;
while(i>1)
{
   i=i % 2;
   x=x+1;
}
```

جواب: $O(1)$

مرتبه لگاریتمی

حلقه زیر را در نظر بگیرید:

```
for ( i=b ; i>=a ; i=i/k )
    s;
```

```
for ( i=a ; i<=b ; i=i*k )
    s;
```

تعداد تکرار : $\log_k^b - \log_k^a + 1$

اگر این مقدار صحیح نبود، حد بالای آن را در نظر بگیرید.

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=1 ; i<=8 ; i=i*2 )
        s;
```

جواب: $\log_2^8 - \log_2^1 + 1 = 4$

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=27 ; i<=n ; i=i*3  )
        s;
```

جواب:

$$\log_3^n - \log_3^{27} + 1 = \log_3^n - 2 \Rightarrow O(\log_3^n)$$

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=n ; i>=16 ; i=i/4  )
        s;
```

جواب:

$$\log_4^n - \log_4^{16} + 1 = \log_4^n - 1 \Rightarrow O(\log_4^n)$$

مثال

```
for ( i=n ;  i>=1;  i=i - i/3 )
        s;
```

```
for ( i=n ;   i>=1;  i= i / (3/2)  )
        s;
```

تعداد تکرار را مشخص کنید.

جواب:

می توان نوشت : $i = \frac{2}{3} i = \frac{i}{\frac{3}{2}}$

بنابراین تعداد تکرار برابر است با :

$$\log_{3/2}^{n} - \log_{3/2}^{1} + 1 = \log_{3/2}^{n} + 1$$

حلقه های تودرتو

تعداد تکرار را مشخص کنید.

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

حل:

تعداد تکرار هر حلقه: n

تعداد تکرار دستور S : n^2

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=2 ; i<=n ; i=i+4 )
    for ( j=n ; j>3 ; j=j-2 )
            s;
```

جواب:

$$\frac{n-2+1}{4} \times \frac{n-3}{2} \Rightarrow O(n^2)$$

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=1 ; i<=n ;  i=i*2 )
      for ( j=1 ; j<=n ; j++ )
                  s;
```

جواب:

$$(\lg n + 1) \times n \Rightarrow O(n \lg n)$$

حلقه های پشت سرهم

تعداد تکرار را مشخص کنید.

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

جواب: $O(\max(n,m))$

یا $O(n+m)$

ترکیب حلقه های تودر تو و پشت سرهم

تعداد تکرار را مشخص کنید.

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

جواب:

$$O(\max(n^2, n)) = O(n^2)$$

حلقه های تودرتو وابسته

تعداد تکرار را مشخص کنید.

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

جواب:

i	1	2	3	...	n
تعداد تکرار	1	2	3	...	n

مجموع تعداد تکرار :

$$1+2+3+\ldots+n=\frac{n(n+1)}{2}\Rightarrow O(n^2)$$

روش دوم:

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

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=1 ; i<=n ; i=i*2 )
    for ( j=1 ; j<=i ; j++ )
            s;
```

جواب:

i	1	2	4	...	n
تعداد تکرار	1	2	4	...	n

مجموع تعداد تکرار :

$$1+2+4+\ldots+n=2^0+2^1+2^2+\ldots+2^{\log_2^n}=\frac{2^{\lg n+1}-1}{2-1}=2^{\lg n}\times 2-1=2n-1\Rightarrow O(n)$$

$$a^0+a^1+a^2+a^3+\ldots+a^k=\frac{a^{k+1}-1}{a-1}$$

مثال

تعداد تکرار را مشخص کنید.

```
for ( k=1 ; k<=n ; k++ )
    for ( i=1 ; i<=n ; i=i*2 )
        for ( j=1 ; j<=i ; j++ )
            s;
```

جواب:

در مثال قبل دیدیم که دو حلقه داخلی (2n-1) مرتبه تکرار می شود. چون حلقه اول n مرتبه تکرار می شود، تعداد کل تکرار برابر است با:

$$n(2n-1) \Rightarrow O(n^2)$$

مثال

تعداد تکرار را مشخص کنید.

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

جواب:

i	1	2	3	...	n
تعداد تکرار	n	n/2	n/3	⋯	1

مجموع تعداد تکرار :

$$n+\frac{n}{2}+\frac{n}{3}+\ldots+1=n(1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n})\approx n\ln n\Rightarrow O(n\lg n)$$

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=1 ; i<=n ; i=i*3 )
    for ( j=i ; j<=n ; j++ )
        s;
```

جواب:

i	1	3	9	...	n
تعداد تکرار	n-1+1	n-3+1	n-9+1	...	n-n+1

مجموع تعداد تکرار :

$$(\log_3^n+1)(n+1)-(1+3+9+\ldots+n)=(\log_3^n+1)(n+1)-\frac{3n-1}{2}\Rightarrow O(n\lg n)$$

$$1+3+9+\ldots+n=3^0+3^1+3^2+\ldots+3^{\log_3^n}=\frac{3^{\log_3^n+1}-1}{3-1}=\frac{3n-1}{2}$$

مثال

تعداد تکرار را مشخص کنید.

```
for ( i=1 ; i<=n ; i++ )
    for ( j=1 ; j<=log(i) ; j++ )
            s;
```

جواب:

i	1	2	...	n
تعداد تکرار	log(1)	log(2)	...	log(n)

مجموع تعداد تکرار :

$$\log(1)+\log(2)+\ldots+\log(n)=\log(1\times 2\times\ldots\times n)=\log(n!)$$

$$\log(n!)=\theta(n\lg n)$$

تغییر مقدار نهایی شمارنده

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

تعداد تکرار را مشخص کنید.

جواب:

i	1	2	3	...
تعداد تکرار	n/2	n/4	n/8	...

مجموع تعداد تکرار :

$$\frac{n}{2}+\frac{n}{4}+\frac{n}{8}+\ldots=n(\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots)=n\times\frac{1/2}{1-1/2}=n\Rightarrow O(n)$$

پیچیدگی در حالت فراخوانی تابع

مثال : با فرض اینکه پیچیدگی زمانی تابع f(k) برابر O(k) باشد، پیچیدگی را مشخص کنید.

```
for ( i=1 ; i<=n ; i++ )
        f(n);
```

جواب: $n \times n \Rightarrow O(n^2)$

مثال :

```
for ( i=1 ; i<=n ; i++ )
        f(i);
```

جواب: $f(1)+f(2)+\ldots+f(n)=1+2+\ldots+n=\frac{n(n+1)}{2}\Rightarrow O(n^2)$

مسئله خرس قطبی

در یک زمستان سرد ، خرس قطبی n قطعه گوشت (دقیقا به اندازه های 1، 2، تا n) را در غاری ذخیره کرده است. او هر روز یکی از این قطعه گوشت ها را به صورت تصادفی انتخاب می کند. اگر اندازه ی گوشت عدد فردی بود، آن را کاملا می خورد. اگر زوج بود، آن را دقیقا نصف می کند، یک نصف آن را می خورد و نصف دیگر را مجددا در غار قرار می دهد. اگر گوشتی موجود نباشد، خرس می میرد. با این الگوریتم، برای n های خیلی بزرگ، روزهای باقیمانده از عمرخرس چگونه خواهد بود؟

حل:

با فرض n=4 ، قطعه های گوشت برابر 1 و2 و3 و4 می باشند. با هر ترتیبی که بخورد بعد از 7 روز گوشت ها تمام می شود. برای n=8 جواب 15 است. جواب این مسئله 2n-1 است.

روش دوم :

$$n+\frac{n}{2}+\frac{n}{4}+\ldots=n(1+\frac{1}{2}+\frac{1}{4}+\ldots)\approx n\times\frac{1}{1-\frac{1}{2}}\approx 2n=O(n)$$

تابع $\lg^* n$

خروجی این تابع برابر است با تعداد دفعاتی که اگر از n لگاریتم گرفته شود، حاصل برابر یک(یا کوچکتر) خواهد شد. بنابراین رشد این تابع بسیار کند است.

$\lg^* 4 = 2$ $\quad 4 \to 2 \to 1$

$\lg^* 16 = 3$ $\quad 16 \to 4 \to 2 \to 1$

$\lg^* 65536 = 4$ $\quad 65536 \to 16 \to 4 \to 2 \to 1$

تعریف نماد اوی کوچک و امگای کوچک

عبارت $f(n)= o(g(n))$ برقرار است در صورتی که : $\lim_{n\to\infty}\frac{f(n)}{g(n)}=0$

عبارت $f(n)= \omega(g(n))$ برقرار است در صورتی که : $\lim_{n\to\infty}\frac{f(n)}{g(n)}=\infty$

مثال: رابطه $\frac{n^2}{2}\in o(n^3)$ برقرار است:

$$\lim_{n\to\infty}\frac{n^2/2}{n^3}=\lim_{n\to\infty}\frac{1}{2n}=0$$

تمرین

تعداد تکرار و مرتبه اجرایی را در هر یک از حلقه های زیر مشخص کنید.

```
for ( i=1 ; i<=n ; i++ )
  for ( j=1 ; j<=i  ; j++)
    for ( k=1; k<=j ; k++)
        s;
```

```
for ( i=1 ; i<n/2 ; i++ )
  for( j=n/2 ;  j<n ; j++)
    for( k=0 ; k<i+j ; k++)
        s;
```

```
for ( i=1 ; i<=n; i++ )
    for ( j=i ; j<=n ; j=j*3 )
            s;
```

```
for ( t=1 ; t<=n-1 ; t++ ) {
    for( i=1 ; i<n-t ; i++) {
        j=i+t;
        for( k=i ; k<=j-1 ; k++)
            s;
} }
```

تمرین

نشان دهید که :

$4^{\lg n} \in \theta(n^2)$ $\quad e^n \in \Omega(n2^n)$ $\quad 3^{2^n} \in O(2^{3^n})$ $\quad \lg^*(n^n) \in O(n)$

$n^{\frac{1}{\lg n}} \in \theta(1)$ $\quad 2^{\sqrt{2\lg n}} \in \Omega(\lg^2 n)$ $\quad \lg^* n \in \theta(\lg^*(\lg n))$ $\quad 4^{\log n} \in \Omega\left(\lg^2 n\right)$

تمرین: کامپیوتری یک مسئله به اندازه ۸ را با الگوریتمی از مرتبه $o(2^n)$ در یک واحد زمان حل می کند. اگر سرعت کامپیوتر ۶۴ برابر شود، چه اندازه ای از همان مسئله را در یک واحد زمان حل خواهد کرد؟

این اسلاید ها بر مبنای نکات مطرح شده در فرادرس
«مجموعه فرادرس های ساختمان داده ها»
تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید.

faradars.org/fvds9402