Function growth

یادآوری

```
Algorithm prefixAverages1(X, n)
    Input array X of n integers
    Output array A of prefix averages of X

  A ← new array of n integers
  for i ← 0 to n – 1 do
      s ← X[0]
      for j ← 1 to i do
          s ← s + X[j]
      A[i] ← s / (i + 1)
  return A
```

```
Algorithm f(n, m)
        k ← m
        for i ← 1 to m do
            j ← 1
            while j < n do
                k ← k + 1
                j ← 2*j
```

یادآوری

Algorithm f(n, m)

    k ← m
    for i ← 1 to m do
        j ← 1
        while j < n do
            k ← k + 1
            j ← 2*j

j:

$$\left.\begin{array}{c} 1 \\ 2 \\ 4 \\ \vdots \end{array}\right\} \lg n$$

$$> n$$

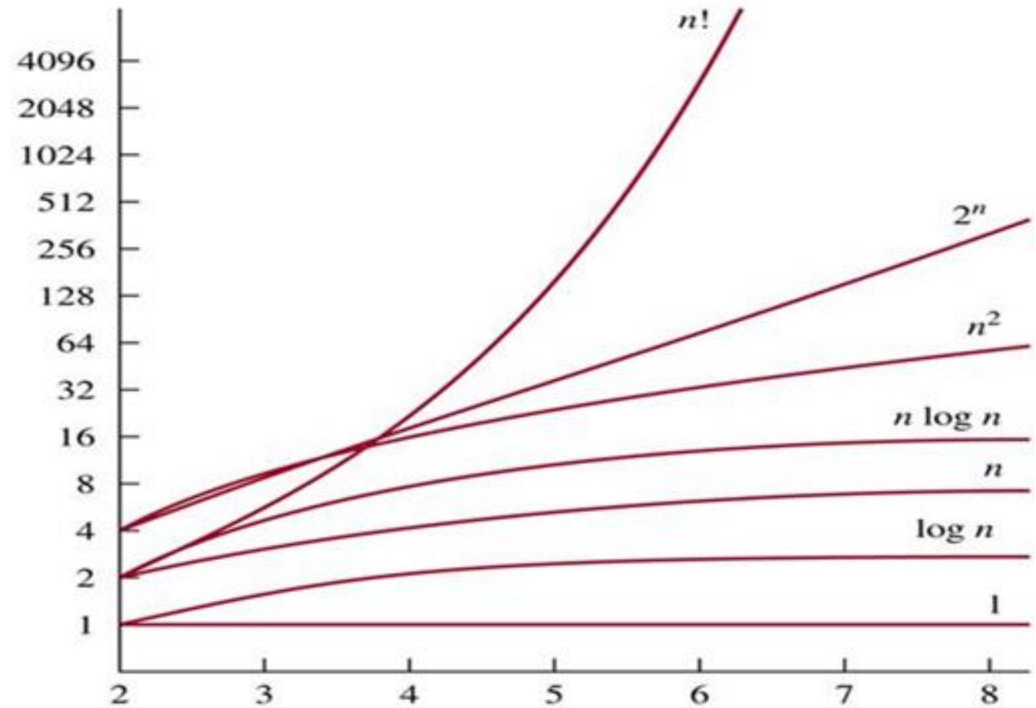
$$2^i = n \Longrightarrow i = \lg n$$

$$\Downarrow$$

$$m \lg n$$

Growth of common functions

- 1
- $\log n$
- n
- $n \log n$
- n^2
- 2^n
- $n!$



A Display of the Growth of Functions Commonly Used in Big-O Estimates

Consumed time of common functions

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} years
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years	very long
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years	very long
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

Notations

$$\Theta(g(n)) = \{ f(n) : \text{ there exist positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0 \} .$$ [1]

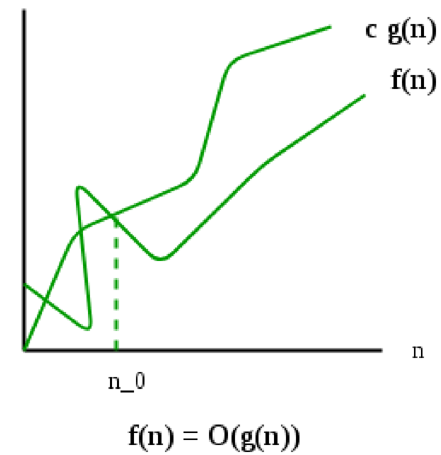
$$O(g(n)) = \{ f(n) : \text{ there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le c g(n) \text{ for all } n \ge n_0 \} .$$

$$\Omega(g(n)) = \{ f(n) : \text{ there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le c g(n) \le f(n) \text{ for all } n \ge n_0 \} .$$

نماد O

- فرض کنید $f(n) = O(g(n))$
- نماد O در واقع یک حد بالای مجانبی برای تابع f(n) مشخص می‌کند.

$$O\big(g(n)\big) = \{f(n) | (\exists\, c, n_0 > 0), \forall n > n_0 : 0 \le f(n) \le cg(n)\}$$



$$T(n) = \frac{1}{3}n^2 - n \Rightarrow O(n^2)$$

$$\frac{1}{3}n^2 - n \le cn^2 \overset{c=\frac{1}{3}}{\Longrightarrow} n > 0 \Rightarrow \Rightarrow n_0 = 0$$

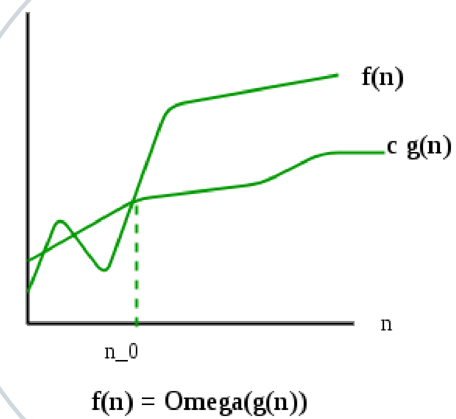
نماد Ω

- فرض کنید $f(n) = \Omega(g(n))$
- نماد Ω در واقع یک حد پایین مجانبی برای تابع f(n) مشخص می‌کند.

$$\Omega\big(g(n)\big) = \{f(n) | (\exists\ c, n_0 > 0), \forall n > n_0 : 0 \le cg(n) \le f(n)\}$$

$$T(n) = \frac{1}{3}n^2 - n \Rightarrow \Omega(n^2)$$

$$cn^2 \le \frac{1}{3}n^2 - n \Longrightarrow \overset{c=\frac{1}{6}}{\Longrightarrow} -\frac{1}{6}n^2 + n \le 0 \Rightarrow n \ge 6 \Rightarrow n_0 = 6$$

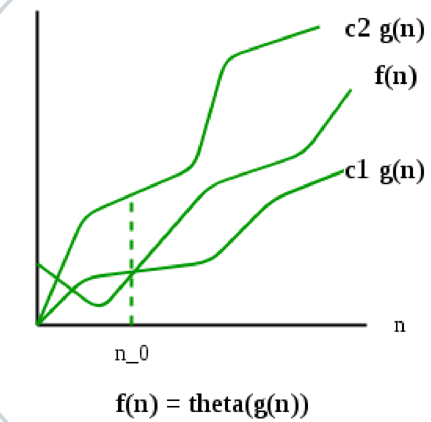


نماد Θ

- فرض کنید $f(n) = \Theta(g(n))$.
- نماد Θ نماینده مجموعه‌ای از توابع است که نماینده‌ی آنها تابع $g(n)$ است:

$$\Theta\big(g(n)\big) = \{f(n) | (\exists c_1, c_2, n_0 > 0), \forall n > n_0 : c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

- $T(n) = 3n^2 + 2n \Rightarrow \Theta(n^2)$
- c1,c2,n0=?



ادامه مثال

- حل:

$$T(n) = 3n^2 + 2n \Rightarrow \Theta(n^2)$$

$$c_1 n^2 \le 3n^2 + 2n \xRightarrow{c1=1} 2n^2 + 2n \ge 0 \Rightarrow$$

$$c_2 n^2 \ge 3n^2 + 2n \xRightarrow{c_2=4} n \ge 2$$

$$c_1 = 1\ ,\ c_2 = 4, n_0 = 2$$

- یک مثال دیگر:

$$T(n) = \frac{1}{3}n^2 - n \Rightarrow \Theta(n^2)$$

$$c_1 n^2 \le \frac{1}{6}n^2 - n \xRightarrow{c1=\frac{1}{6}} -\frac{1}{6}n^2 + n \le 0 \Rightarrow n \ge 6$$

$$c_2 n^2 \ge \frac{1}{3}n^2 - n \xRightarrow{c_2=\frac{1}{3}} n > 0$$

$$c_1 = \frac{1}{6},\ c_2 = \frac{1}{3}, n_0 = 6$$

Notations

For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

$$f(n) = O(g(n)) \quad \text{is like} \quad a \leq b\,,$$
$$f(n) = \Omega(g(n)) \quad \text{is like} \quad a \geq b\,,$$
$$f(n) = \Theta(g(n)) \quad \text{is like} \quad a = b\,.$$

چند مثال

- یک قانون ساده برای توابع رشد چندجمله‌ای:
  - تمامی مقادیر ثابت (چه به عنوان ضریب و چه در جمع یا تفریق) و همچنین جملات مرتبه پایین‌تر قابل حذف هستند.
  - این قانون برای Ω و O هردو برقرار است.
- مثال:
  - مرتبه $50n^3 + 100n + 11$ چیست؟
  - مرتبه $4n^2$ چیست؟
    - $O(n^2)$؟
    - $\Omega(n^2)$؟
    - $O(n^3)$؟
    - $\Omega(n^3)$؟

$$4n^2 > cn^3 \overset{n\neq 0}{\Longrightarrow} cn < 4$$

- که برای هیچ مقداری از c نمی‌تواند شرایط مورد نظر ما را برقرار کند.

چند مثال

- ✓ مرتبه $n^2 - n$ چطور؟
  - روشن است که $\Omega(n^2)$ است.
  - $\Omega(n^2)$ چطور؟

$$n^2 - n < cn^2 \overset{n\neq 0}{\Longrightarrow} n - 1 < cn \Rightarrow n(c-1) > -1 \overset{c\neq 1}{\Longrightarrow} n > \frac{1}{c-1}$$

  - کافی است برای یک مقدار c درست باشد. که برای تمام مقادیر نامساوی ۱ برقرار است.

- ✓ 3log(n) + log (log (n)) = O(?)

More examples and proof

- $3n^2 + 2n = f(n) = O(n^2)$
  - ✓ $3n^2 + 2n \le cn^2$
  - ✓ $c = 4 \Rightarrow n^2 - 2n \ge 0 \Rightarrow n \ge 2 \Rightarrow n_0 = 2$

- $3n^2 + 2n = f(n) = \Omega(n^2)$
  - ✓ $cn^2 \le 3n^2 + 2n$
  - ✓ $c = 2 \Rightarrow -n^2 - 2n \le 0 \Rightarrow n \ge -2 \Rightarrow n_0: all\ numbers\ greater\ tahn\ 0$
- $\frac{1}{3}n^2 - n = O(n^2)$
  - ✓ $\frac{1}{3}n^2 - n \le cn^2$
  - ✓ $c = 4 \Rightarrow \frac{2}{3}n^2 + n \ge 0 \Rightarrow n \ge -\frac{3}{2} \Rightarrow n_0: all\ numbers\ greater\ tahn\ 0$

چند قانون ساده دیگر:

- If $f_1(n) = O(g_1(n))$ *and* $f_2(n) = O(g_2(n))$
  - ✓ $f_1(n) + f_2(n) = O(g_1(n) + g_2(n)) = O(\max(g_1(n), g_2(n)))$
  - ✓ $f_1(n) * f_2(n) = O(g_1(n) * g_2(n))$
- $\log n^x$ is $O(\log n)$, for $x > 0$
- $\log n \neq \Omega(n^x)$, for $x > 0$
  - ✓ یعنی رشد n برای هیچ توان مثبتی از آن، کوچکتر از رشد $\log n$ نیست.
- $\log^x n$ is $O(n^y)$ for $x > 0$ and $y > 0$

Inappropriate Expressions

$$f(n) \cancel{\leq} O(g(n))$$

$$f(n) \cancel{\geq} O(g(n))$$

Asymptotic analysis - terminology

- Special classes of algorithms:
  - ***logarithmic***: **O(log n)**
  - ***linear***: **O(n)**
  - ***quadratic***: **$O(n^2)$**
  - ***polynomial***: **$O(n^k)$, $k \geq 1$**
  - ***exponential***: **$O(a^n)$, $n > 1$**
- Polynomial vs. exponential ?
- Logarithmic vs. polynomial ?

نمادهای مجانبی کوچک

نماد ω

وقتی می‌گوئیم f(n)=ω(g(n))

یعنی مرتبه g(n) اینقدر از f(n) کوچک‌تر هست که:

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=\infty$$

نماد o

وقتی می‌گوئیم f(n)=o(g(n))

یعنی مرتبه f(n) اینقدر از g(n) کوچک‌تر هست که:

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=0$$