Examples of Using Big-O

1. To which big-O class belongs the following functions:

a. $f(n) = 2n+5, \quad n \geq 0$

Solution:

$f(n) = 2n+5 \leq 2n+n = 3n$ for all $n \geq 5$

Let $c = 3$, $n_0 = 5$ and $g(n) = n$, then

$f(n) \leq c \cdot g(n)$ for all $n \geq n$.

it now follows that $f(n) = O(g(n)) = O(n)$

b. $f(n) = n^2 + 3n + 2, \quad g(n) = n^2, \quad n \geq 0$.

Solution:

Note that $3n+2 \leq n^2$ for all $n > 4 \Rightarrow$

$f(n) = n^2 + 3n + 2 \leq n^2 + n^2 \leq 2n^2 = 2 \cdot g(n)$ for all $n > 4$

Let $c = 2$ and $n_0 = 4$ then

$f(n) \leq c\, g(n)$ for all $n \geq 4 \Rightarrow$

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

c. $f(n) = a$, where a is a nonnegative real number and $n \geq 0$.,

f is a constant function.

$f(n) = a \leq a \cdot 1$ for all $n \geq a$

4 $c = a$, $n_0 = a$ and $g(n) = 1 \Rightarrow$

$f(n) = O(g(n)) = O(1)$.

d. $T(n) = 15n^3 + 7n^2 + 35$

Solution:

$T(n) = 15n^3 + 7n^2 + 35 \leq 57n^3 \leq 57 \cdot g(n)$

$c_1 = 57$, $n_0 = 1 \Rightarrow$

$T(n) \leq 57\, g(n)$, for all $n > 1$.

$T(n) = O(g(n)) = O(n^3)$.

e. $f(n) = 10n^3 + n \log n$

Solution:

$f(n) \leq 11n^3 \leq 11 \cdot g(n) \leq c \cdot g(n)$

$c = 11$, $n_0 = 1$

② Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$ ②

for $x > 1$ we have

$$x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2$$

$$\Rightarrow \quad x^2 + 2x + 1 \leq 4x^2$$

Therefore, for $c = 4$ and $k = 1$:

$f(x) \leq c\,x^2$ whenever $x > k$

$\Rightarrow$ $f(x)$ is $O(x^2)$.

Note: if $f(x)$ is $O(x^2)$, it is also $O(x^3)$?

Yes. x^3 grows faster than x^2, so x^3 grows also faster than $f(x)$.

Therefore, we always have to find smallest simple function $g(x)$ for which $f(x)$ is $O(g(x))$

Useful Rules for Big-O

1. for any polynomial $f(n)$ (nonnegative real-valued function) such that

$$f(n) = a_m \cdot n^m + a_{m-1} \cdot n^{m-1} + \cdots + a_1 \cdot n + a_0$$

where a_i's are real numbers, $a_m \neq 0$, $n \geq 0$, and m is nonnegative integer, Then

$$f(n) = O(n^m)$$

examples: in the following, $f(n)$ is a nonnegative real-valued function.

- $f(n) = an + b$, where a and b are real numbers and a is nonzero.

  Answer: $f(n) = O(n)$.

- $f(n) = n^2 + 2n + 7$

  Answer: $f(n) = O(n^2)$.

- $f(n) = 15n^3 + 10n^{15}$

  Answer: $f(n) = O(n^{15})$

2. if $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$,

then $(f_1 + f_2)(n)$ is $O(\max(g_1(n), g_2(n)))$

3. if $f_1(n)$ is $O(g(n))$ and $f_2(n)$ is $O(g(n))$, then

$$(f_1 + f_2)(n) \text{ is } O(g(n)).$$

4. if $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$ then

$(f_1 \cdot f_2)(n)$ is $O(g_1(n) \cdot g_2(n))$

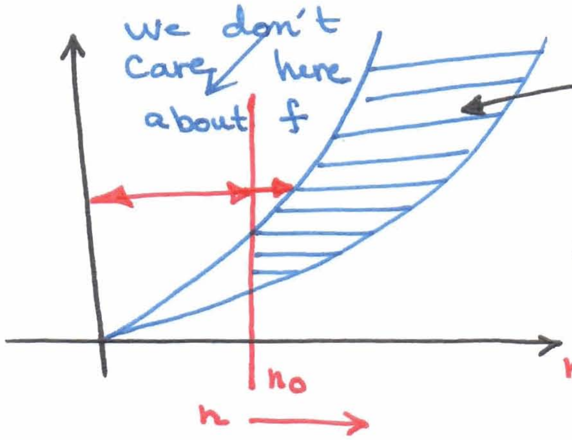
other notations:

1. Θ notation :

f, g : nonnegative functions of nonnegative arguments :

$f(n) = \Theta(g(n))$, if

$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$, for all $n \geq n_0$



this is the region of $f(n)$, that describes by $\Theta(g(n))$

* The most important: to find c_1, c_2 and n_0

Example :

1. $f(n) = 10n^3 + 5n^2 + 17 \in \Theta(n^3)$
2. $f(n) = 2n^3 + 3n + 79 \in \Theta(n^3)$

proof of 1 :

$10n^3 \leq f(n) \leq (10 + 5 + 17)n^3 = 32n^3$

$10n^3 \leq f(n) \leq 32n^3$

$c_1 = 10$, $c_2 = 32$

$c_1 \cdot n^3 \leq f(n) \leq c_2 \cdot n^3$

$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$, for all $n \geq 1$ $\boxed{n_0 = 1}$

3. $f(n) = 10n^3 + n\log n \in \Theta(n^3)$

proof :

$10n^3 \leq f(n) \leq 11n^3$

$c_1 = 10$, $c_2 = 11$, $n_0 = 1$, $g(n) = n^3$.

then $f(n) = \Theta(n^3)$

Θ notation properties:

$f \in \Theta(g_1)$, $h \in \Theta(g_2)$, then

$f + h \in \Theta(g_1 + g_2)$

if $g_1 = g_2 \Rightarrow f + h \in \Theta(g)$

2. Ω notation

f, g: nonnegative functions of nonnegative arguments:

$f(n) = \Omega(g(n))$, if

$c_1 \cdot g(n) \leq f(n)$ for $n > n_0$

homework 1: from textbook (D.S. Malik).

1. Chapter 1: (exercises: page 51)

1.7, 1.8, 1.10, 1.12, 1.13

2. find the Big-O for the following code:

```
for (int i=0; i<m; i++)
    for( int j=0; j<n; j++)
        cout << i*j <<endl;
```