Complexity

Jan 10

1. Running Time of Algs:

Suppose we were to count each step. We can do:

1. read, write vars: 1 each
2. method call: 1 + steps to evaluate each arg + steps to execute method
3. return statement: 1 + steps to eval return value
4. if statement, while statement (Not the entire loop): 1 + Steps to eval exit condition
5. assignment statement: 1 + steps to eval both sides
6. arithmetic, comparison, boolean operators: 1 + steps to eval each operand
7. array access: 1 + steps to eval index
8. constants: Free

E.g. Consider this function for insertion sort.

Pre-cond: A is an array of ints

Post-cond: A is sorted in non-decreasing order

def IS(A):	Steps
`i=1`	2
`while(i < len(A))`	5
`    t = A[i];`	4
`    j = i;`	3
`    while(j > 0 AND A[j-1] > t)`	9
`        A[j] = A[j-1];`	6
`        j = j-1;`	4
`    A[j] = t;`	4
`    i = i+1;`	4

To find its time complexity, suppose A has n elements. I.e. len(A) = n

Line 1 takes 2 steps.
The outer loop (lines 2, 3, 4, 8, 9) runs n-1 times. This is because i starts at 1 and goes to n-1. Therefore, the outer loop takes 20(n-1) steps.
However, line 2 is evaluated one last time, and it takes 5 steps.
Each time the inner loop (lines 5-7) runs, j goes from i to 1. Therefore, it takes 19i steps. However, line 5 is executed once more, which takes 9 steps.

In total, the inner loop takes

$$\sum_{i=1}^{n-1} 19i+9 = \sum_{i=1}^{n-1} 19i + \sum_{i=1}^{n-1} 9$$

$$= \frac{19(n-1)(n)}{2} + 9(n-1)$$

$$= \frac{19n^2}{2} - \frac{19n}{2} + 9n - 9$$

$$= \frac{19n^2}{2} - \frac{n}{2} - 9$$

$$\text{Total} = 2 + 20(n-1) + 5 + \frac{19n^2}{2} - \frac{n}{2} - 9$$

$$= \frac{19n^2}{2} + \frac{31n}{2} - 22$$

However, if I were to run the code on an older computer, it would take more time.

We say that quadratic polynomials are of order n^2.
We say that cubic polys are of order n^3.
We say that $4n\log(n)+2n+10$ is of order $n\log(n)$.

E.g. Show that $12n^2+10n+10$ is of order n^2.

Soln:

$$12n^2+10n+10 \leq 12n^2+10n^2+10$$
$$= 22n^2+10$$

For all $n \geq 10$:

$$22n^2+10 \leq 22n^2+n$$
$$\leq 22n^2+n^2$$
$$\leq 23n^2$$

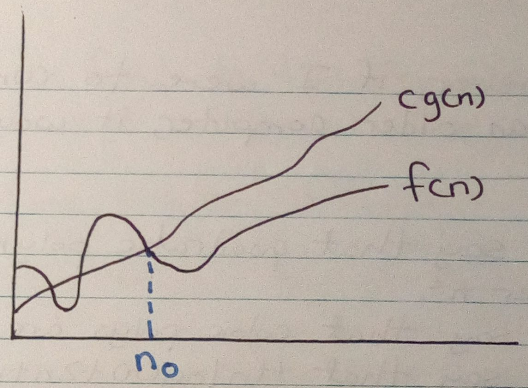
2. Big-O

- Let R^+ be the set of positive, real numbers
- Let R_0^+ be the set of positive, real numbers ≥ 0
- Let N_k be the set of natural numbers $\geq k$
- Let F be the set of functions of $f: N_k \rightarrow R_0^+$.

The defn of Big-O is: Let $g \in F$. $O(g)$ is the set of functions $f \in F$ s.t. $\exists c \in R^+, \exists n_0 \in N, \forall n \in N, n \geq n_0 \rightarrow f(n) \leq c \cdot g(n)$.

$0 \leq$

Hilroy

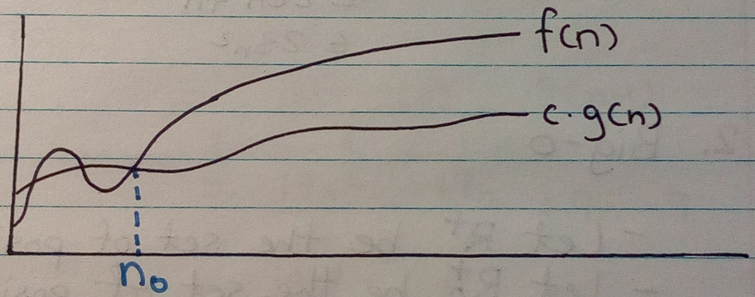


The pic above shows $f(n) = O(g(n))$

Big-O provides an upper bound.

3. Big-Omega (Ω)

Def: Let $g \in F$. $\Omega(g)$ is the set of functions $f \in F$ s.t. $\exists b \in R^+, \exists n_0 \in N, \forall n \in N, n \geq n_0 \longrightarrow f(n) \geq b \cdot g(n) \geq 0$.



The pic above shows $f(n) = \Omega(g(n))$

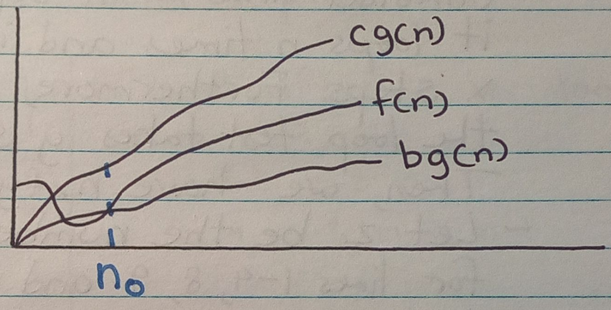
Big-Omega provides a lower bound.

4. Big-Theta (Θ)

Informal def: If $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$, then $f(n) \in \Theta(g(n))$.

Formal def: Let $g \in \mathcal{F}$. $\Theta(g)$ is the set of functions $f \in \mathcal{F}$ s.t. $\exists b \in \mathbb{R}^+$, $\exists c \in \mathbb{R}^+$, $\exists n_0 \in \mathbb{N}$, $\forall n \in \mathbb{N}$, $n \geq n_0 \rightarrow bg(n) \leq f(n) \leq cg(n)$.

Thm: $f(n) \in \Theta(g(n))$ iff $f(n) \in O(g(n))$ and $g(n) \in O(f(n))$.



The above picture shows $f(n) = \Theta(g(n))$.

Big-Theta provides a tight bound.

E.g. Show $12n^2 + 10n + 10 \in \Theta(n^2)$

Soln:

For all $n \geq 10$: $n^2 \leq 12n^2 + 10n + 10 \leq 23n^2$

$\therefore 12n^2 + 10n + 10 \in \Theta(n^2)$

5. Examples

a) Consider the insertion sort function. Prove that it is $\Theta(n^2)$.

Soln:
To prove that it's $\Theta(n^2)$, we need to prove that it's both $O(n^2)$ and $\Omega(n^2)$.

Part 1: $O(n^2)$

- Recall that Big-O provides an upper bound.
- Consider lines 5-7. Suppose that it loops n times and it takes x steps. Furthermore, suppose the loop test takes y steps. Then, we have $nx+y$
- Let z be the number of steps for lines 1-4, 8, 9 and the loop test.

- Lines 2-9 loop at most n-1 times.
  $\therefore$ The function takes $n(nx+y+z)$ steps at most.

$$n(nx+y+z) = xn^2 + n(y+z) \in O(n^2)$$

Note: We overcounted/overestimated the number of times both the inner and outer loops loop.

Part 2: $\Omega(n^2)$

- Recall that Big-Omega provides a lower bound.

- Consider the input that forces the greatest number of steps. It is an array of length n that is sorted in decreasing order.

  I.e. $[n-1, n-2, \ldots, 2, 1, 0]$

- Suppose we ran the function using the list above and we will count 1 per line.

i	A[0...i] after outer loop	Inner loop steps
1	$n-2, n-1$	loops once; $\geq 3\cdot1+1$
2	$n-3, n-2, n-1$	loops twice; $\geq 3\cdot2+1$
3	$n-4, n-3, n-2, n-1$	loops thrice; $\geq 3\cdot3+1$
$\vdots$	$\vdots$	$\vdots$
k	$n-k-1, n-k, \ldots, n-1$	loops k; $\geq 3\cdot k+1$
$\vdots$	$\vdots$	$\vdots$
$n-1$	$0, 1, 2, \ldots, n-1$	$\geq 3(n-1)+1$

Therefore, the function takes

$$\sum_{i=1}^{n-1} 3i+1 \text{ steps.}$$

$$\sum_{i=1}^{n-1} 3i+1 = \frac{3}{2}n^2 - \frac{n}{2} - 1$$

$$\in \Omega(n^2)$$

Therefore, the function $\in \Theta(n^2)$.

b) Prove $n^3 - n^2 + 5 \in \Theta(n^3)$

Soln

We have to prove $n^3 - n^2 + 5 \in O(n^3)$ and $n^3 - n^2 + 5 \in \Omega(n^3)$.

To prove $n^3 - n^2 + 5 \in O(n^3)$:

$$n^3 - n^2 + 5 \leq n^3 + 5$$
$$\leq n^3 + 5n^3$$

When $n \geq 2$, $n^3 + 5n^3 = 6n^3$

Let $n_0 = 1$ and $c = 6$. Then, $n^3 - n^2 + 5 \in O(n^3)$

To prove $n^3 - n^2 + 5 \in \Omega(n^3)$:

$$n^3 - n^2 + 5 > n^3 - n^2$$
$$\geq bn^3$$

Dividing both sides by n^2, we get:

$$n - 1 \geq bn$$
$$n - bn \geq 1$$
$$n \geq \frac{1}{1-b}$$

Since we want $n \geq n_0$, where $n_0 \in \mathbb{N}$, we should pick $b < 1$.

Let $b = \frac{1}{2}$. Then, $n = 2$, and $n_0 = 2$.

6. Using Limits to Prove/Disprove Big-O

Assume: $\exists n_0 : \forall n \geq n_0 : f(n) \geq 0$ and $g(n) \geq 0$

Thm: If $\lim_{n \to \infty} \frac{f(n)}{g(n)}$ exists and is finite, then $f(n) \in O(g(n))$

E.g. Prove $\frac{n(n+1)}{2} \in O(n^2)$

Soln:

$$\lim_{n \to \infty} \frac{n(n+1)}{2n^2}$$

$$= \lim_{n \to \infty} \frac{n^2+n}{2n^2}$$

$$= \frac{1}{2} \lim_{n \to \infty} \frac{n^2+n}{n^2}$$

$$= \frac{1}{2}\left(\lim_{n \to \infty} \frac{n^2}{n^2} + \lim_{n \to \infty} \frac{n}{n^2}\right)$$

$$= \frac{1}{2}(1+0)$$

$$= \frac{1}{2}$$

E.g. Prove $\ln(n) \in O(n)$

Soln:

$$\lim_{n \to \infty} \frac{\ln(n)}{n}$$

$$= \lim_{n \to \infty} \frac{1}{n} \leftarrow \text{l'hopital's rule}$$

$$= 0$$

Hilroy

Thm: If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, then $f(n) \notin O(g(n))$.

E.g. Disprove $n^2 \in O(n)$

Soln:

$$\lim_{n\to\infty} \frac{n^2}{n}$$
$$= \lim_{n\to\infty} n$$
$$= \infty$$

E.g. Disprove $n \in O(\ln(n))$

Soln:

$$\lim_{n\to\infty} \frac{n}{\ln(n)}$$
$$= \lim_{n\to\infty} \frac{1}{1/n}$$
$$= \lim_{n\to\infty} n$$
$$= \infty$$

Thm: If $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ DNE and is not ∞, then there is no conclusion.

This happens with piece-wise functions.

7. How can Big-O be abused?

- Consider the 2 statements below:
  1. $10^{100} n \in O(n)$
  2. $n + 10000 \in O(n)$

  These are not practical algo times, but O, Θ can't detect them. This is a price for ignoring machine differences. These pathological cases are rare. O and Θ are usually informative.

Supplemental Big-O / Big-Omega / Big-Theta Notes

- Given a function, to find its time complexity in Big-O, we overestimate the steps it takes. To find its time complexity in Big-Omega, we find the worst input and underestimate the steps it takes.

- $f \in O(g)$ and $f \in \Omega(g) \Leftrightarrow f \in \Theta(g)$
  $f \in O(g) \Leftrightarrow g \in \Omega(f)$

- If $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in O(h(n))$.

- If $\lim_{n \to \infty} \frac{f(n)}{g(n)} \in [0, \infty)$, then $f(n) \in O(g(n))$

  If $\lim_{n \to \infty} \frac{f(n)}{g(n)} \in (0, \infty]$, then $f(n) \in \Omega(g(n))$.

If $\lim_{n \to \infty} \frac{f(n)}{g(n)} \in (0, \infty)$, then $f(n) \in \Theta(g(n))$.

- Log Rules:
  1. $\log_b(xy) = \log_b x + \log_b y$
  2. $\log_b\left(\frac{x}{y}\right) = \log_b x - \log_b y$
  3. $\log_b(x^y) = y \cdot \log_b x$
  4. $\log_b x = \frac{1}{\log_x b}$
  5. $\log_b x = \frac{\log_c x}{\log_c b}$

- Exponent Rules:
  1. $a^n \cdot a^m = a^{n+m}$
  2. $\frac{a^n}{a^m} = a^{n-m}$
  3. $a^n \cdot b^n = (a \cdot b)^n$
  4. $\frac{a^n}{b^n} = \left(\frac{a}{b}\right)^n$
  5. $(b^n)^m = b^{n \cdot m}$

- E.g.
  1. Show that $6n^5 - n^3 + n^2 \in \Theta(n^5)$

     Soln:

     1. Big-O:

     $6n^5 - n^3 + n^2 \leq 6n^5 + n^2$

     $\leq 6n^5 + n^5, \quad n \geq 1$

     $= 7n^5$

     $\therefore C = 7, \ n_0 = 1$

2. Big-Omega:

$$6n^5 - n^3 + n^2 \geq 6n^5 - n^3$$
$$\geq 6n^5 - n^5, \quad n \geq 1$$
$$= 5n^5$$
$$\therefore c = 5, \; n_0 = 1$$

2. Prove $n^2 + 42n + 7 \in O(n^2)$

Soln:

1. Def:

$$n^2 + 42n + 7 \leq n^2 + 42n^2 + 7n^2, \quad n \geq 1$$
$$= 50n^2$$
$$\therefore c = 50, \; n_0 = 1$$

2. Limit:

$$\lim_{n \to \infty} \frac{n^2 + 42n + 7}{n^2}$$
$$= \lim_{n \to \infty} 1$$
$$= 1$$
$$\therefore n^2 + 42n + 7 \in O(n^2)$$

3. Prove $5n\log_2 n + 8n - 200 \in O(n\log_2 n)$

Soln:

$$5n\log_2 n + 8n - 200 \leq 5n\log_2 n + 8n$$
$$\leq 5n\log_2 n + 8n\log_2 n, \quad n \geq 2$$
$$= 13n\log_2 n$$
$$\therefore c = 13, \; n_0 = 2$$