CSE221 ALGORITHMS

Topic: Time Complexity,
Upper Bound
Asymptotic, Lower Bound

Prepared by:
Saad Bin Sohan

BRAC University

Email: sohan.academics@gmail.com

GitHub: https://github.com/saad-bin-sohan

বি. দ্র: নিচের ফাংশনগুলোকে don't think of them as output of x such as $f(x)$

think them as:

"how many times the codes will execute"

Topic-1: Time Complexity

Types of functions

① $f(n) = \underbrace{\text{constant}}_{10,\,20,\,30\ \text{etc...}}$

like $f(n) = 10$

no matter what the input is, how many times the code will run, will always be same and constant

② $f(n) = \log n$ → logarithmic

[উদাহরণ: $f(100) = \log_{10}100 = \log_{10}10^2$

[redacted]

$$= 2\log_{10} 10$$

$$= 2 \times 1 \quad [\because \log_b b = 1]$$

$$= 2$$

but from now, the base will be 2.
(not 10 or e)]

so now,

$$f(16) = \log_2 16 = \log_2 2^4 = 4\log_2 2$$

$$= 4 \times 1$$

$$= 4$$

③ $f(n) = n \rightarrow$ linear

number of times the code will execute $(n) \propto$ input (n)

④ $f(n) = n \log n$

⑤ $f(n) = n^2$ → quadratic

⑥ $f(n) = n^3$ → cubic

⑦ $f(n) = 2^n$ → exponential

Function add

$$\underbrace{\underset{\text{quadratic}}{3n^2} + \underset{\text{linear}}{5n} + \underset{\text{constant}}{10}}$$

as a whole, we will call this a Quadratic function

☆ Sub function গুলোর মধ্যে যার complexity সবচে বেশি, তার নামে function এর নাম

Time complexity = Function এর complexity বা strength যাচাই

এখানে কোনো serial function কে নিচ্ছে, ঐ function কত বেশি complex

Function এর strength যাচাই:

same

input এর জন্য একাধিক function এর মধ্যে যার output যত বড় সে তত stronger (means তার complexity তত বেশি).

যেমন: $f(n) = \log_2 n$ এবং $f(n) = n$, কে বেশি complex?

Ans: let $n = 16$

$$f(n) = \log_2 16 = \log_2 2^4 = 4$$

$$f(n) = n$$

$$f(16) = 16$$

বেশি complex

Time complexity: Time complexity of an algorithm tells us how severely the execution time required to run a program increases or decreases as its input size increases or decreases.

Time Complexity

→ it means function বা algorithm এর runtime এর (kind of) হিসাব

Time Complexity

- Asymptotic time complexity
  - → upper bound
  - → lower bound
  - → Tight bound
- Recursive time complexity
- Iterative time complexity

☆ Asymptotic time complexity tells you how execution time increases/ decreases as you change input size.

e.g. if a program has time complexity of $O(n^2)$, it means the execution time will grow quadratically as the input size n increases.

\$\$\$ for this particular $O(n^2)$,

execution time $\propto n^2 \times$ input size

☆☆☆ Upper bound asymptotic time complexity:

VVI →

→ strength তোলার / same

যেমন: example of such statements:

✓ n^3 is the upper bound of $3n^2$ (True)

✓ n^2 is the upper bound of n (True)

× n is the upper bound of $3n^2+5$ (False)

the task is to find the truthness of the statement

কোনো প্রোগ্রামের upper bound হলো any program that has higher or similar time complexity than it.

এবং কোনো algorithm এর worst case scenario তে program টি তার time complexity এর সর্বোচ্চ মান use করলে তা প্রোগ্রামটি সর্বোচ্চ সংখ্যক বার চলতো।

এজন্য worst case scenario-র time complexity ও একটা upper bound of that given program.

Que Verify, $3n^2 = O(n^3)$

big Oh (symbol of upper bound)

soln:

step-1: given equation কে sentence form দেওয়া

1) $\underbrace{n^3}$ is the upper bound of $\underbrace{3n^2}$

big Oh এর মধ্যে/সমান সমানে right side এ

left side এর function

[redacted]

step-2: statement এর respect এ truthness সত্য/ মিথ্যা claim কর।

2) True

step-3: to prove your claim

[for any question, prove $f(n) = O(g(n))$,

Find two positive constants c and n_0

where $f(n) \le c * g(n)$ for all $n > n_0$.

And $\underbrace{c > 0 \quad \text{and} \quad n_0 \ge 1}$

all real numbers in range (rational and irrational)

[redacted]

3) $3n^2 \le c\,n^3$

$\boxed{\begin{array}{l} c = 1 \\ n_0 = \end{array}}$

for $c = 1$ (any random value that is greater than zero)

it gives us,

$3n^2 \le n^3$

let $n = 1$ $\Rightarrow$ $3n^2 \le n^3$

$3(1)^2 \le 1^3$

$3 \le 1$ (False)

again, $n = 2$

$\Rightarrow 3n^2 \le n^3$

$3(2)^2 \le 2^3$

$12 \le 8$ (False)

again, $n = 3$

$$\Rightarrow 3n^2 \leq n^3$$

$$\Rightarrow 3(3)^2 \leq 3^3$$

$$\Rightarrow 27 \leq 27$$ (True)

again,

$n = 4$

$$\Rightarrow 3n^2 \leq n^3$$

$$\Rightarrow 3(4)^2 \leq 4^3$$

$$48 \leq 64$$ (True)

so, $n_0 = 3$

[কারণ, n_0 হলো n এর ঐ value যার থেকে বড় সব value এর জন্য statement টা True. Not necessary যে n_0 কে n (n যদি একটা set হয়) এর least value হতে হবে।]

so, n_0 in the previous example can also be 4.

NB: c এর value usually বড় নেওয়া better

[redacted]

Que Verify,

$$2n^2+3n+5 = O(n^2)$$

$\boxed{\begin{array}{l} c = 10 \\ n_0 = 1 \end{array}}$

soln:

step-1: n^2 is the upper bound of

$2n^2+3n+5$

step-2: I claim the statement to be true.

True.

step-3:

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

let,

$$2n^2 \le 2n^2$$

$$2n^2 + 3n \le 2n^2 + 3n^2$$

$$2n^2 + 3n + 5 \le 2n^2 + 3n^2 + 5n^2$$

$$\Rightarrow 2n^2 + 3n + 5 \le 10n^2$$

[এখানে $2n^2 \le 2n^2$ দিয়ে শুরু করে given equation left side এ term গুলো add করা হইসে and <= এই symbol কে true hold করার জন্য right side এ বেশি power এর term add করা হইসে।

so here's the shortcut

$$2n^2 + 3n + 5 \le cn^2$$

$2+3+5=10=c$ (a good value for c)

basically coefficient গুলো add করা।

now, for $c = 10$,

$$2n^2 + 3n + 5 \leq 10n^2$$

now to get n,

let $n = 1$,

$$2 + 3 + 5 \rightarrow 10 \leq 10 \text{ (True)}$$

let $n = 2$,

$$2(2)^2 + 3(2) + 5 \leq 10(2)^2$$

$$\Rightarrow 19 \leq 40 \text{ (True)}$$

as n will increase it will hold true for all n.

so, lets take $n = 1$.

— x — [redacted]

NB: invalid statement এর proof লাগবে না। just claim False

[redacted]

কোনো প্রোগ্রামের lower bound হলো any program that has lower or similar time complexity as it.

এবং কোনো algorithm এর best case scenario তে program টি তার time complexity এর সর্বনিম্ন মান use করে যা প্রোগ্রামটি সর্বনিম্ন সংখ্যক বার চলবে।

এজন্য best case scenario-র time complexity ও একটা lower bound of that given program since it has a similar or lower time complexity

Lower Bound

lower bound symbol: Ω (Omega)

Steps for $f(n) = \Omega(g(n))$: Find two positive constants c and n_0 where $f(n) \geq c * g(n)$ for all $n \geq n_0$

Que Verify,

$$2n^2 + 3n + 5 = \Omega(n)$$

soln:

(i) n is the lower bound of $2n^2+3n+5$

(ii) True

(iii) $f(n) \geq c * g(n)$

now,

$$\Rightarrow 2n^2+3n+5 \geq cn$$

let $c=1$,

so,

$$2n^2+3n+5 \geq n$$

$n=1$ $\cancel{2(n)}$, $10 \geq 1$ (True)

$n=2$ $\cancel{2(n)}$, $19 \geq 1$ (True)

so,

$$\boxed{\begin{aligned} c &= 1 \\ n_0 &= 1 \end{aligned}}$$