

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>

Ex. 5: নিচের ইন্সট্রাকশন 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 $f(n) = \text{constant}$

10, 20, 30 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 $f(n) = \log n \rightarrow \text{logarithmic}$

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

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

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

$$= 2$$

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

so now,

$$\begin{aligned} f(16) &= \log_2 16 = \log_2 2^4 = 4 \log_2 2 \\ &= 4 \times 1 \\ &= 4 \end{aligned}$$

③ $f(n) = n \rightarrow \text{linear}$

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

④ $f(n) = n \log n$

⑤ $f(n) = n^2 \rightarrow \text{quadratic}$

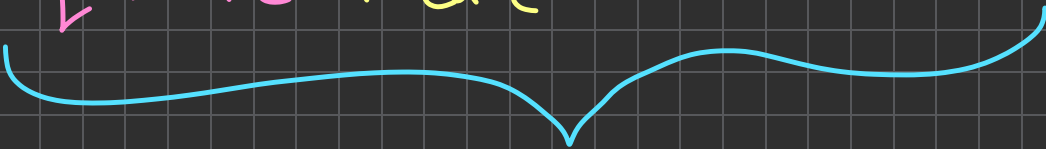
⑥ $f(n) = n^3 \rightarrow \text{cubic}$

⑦ $f(n) = 2^n \rightarrow \text{exponential}$

Function add

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

quadratic linear constant



as a whole, we
will call this a

Quadratic function

★ Sub function સૂત્રો મર્યાદામાં complexity
મર્યાદામાં લેશે, તો નામ function નું નામ

Time complexity = Function এর complexity
বা strength যাতে

এবার কেন এই serial function যা আছে,
এ function তা হোক complex

Function এর strength যাতে:

same

input এর জন্য একটি function এর মধ্যে
যা output যা হতে পারে তাই stronger
(means তা complexity তা হোক).

উদা: $f(n) = \log_2 n$ এর $f(n) = n$, তা হোক complex?

Ans: let $n = 16$

$$\begin{aligned} f(n) &= \log_2 16 \\ &= \log_2 2^4 \\ &= 4 \end{aligned}$$

$$f(n) = n$$

$$f(16) = 16$$

হোক 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 of algorithm w.r. to runtime w.r. (kind of) ~~time~~

Time Complexity

Asymptotic
time
complexity

Recursive
time
complexity

Iterative
time
complexity

→ upper bound
→ lower bound
→ Tight bound

★ 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

IVI \rightarrow time complexity:

\rightarrow strength total / same

□□□: 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

કારણ રજાડામાં upper bound 270
any program that has higher
or similar time complexity than it.

અર કારણ algorithm નો worst case
scenario નો program ને જો તો time
complexity નો અંદાજ માત્ર use કરો
ઠા રજાડામાં અંદાજ અંગ્રજી થોડો બનશે.

બીજા worst case scenario-નો time
complexity 3 વર્ગે upper bound
of that given program.

Que

verify, $3n^2 = O(n^3)$

big Oh (symbol
of upper bound)

Solⁿ:

step-1: given equation in sentence form
[in Hindi]

1) n^3 is the upper bound of $3n^2$

big Oh \Rightarrow

महि/समान समान

right side \Rightarrow

left

side \Rightarrow function

step-2: statement w^o respect to
truthness ~~miss~~ / ~~forget~~ claim ~~not~~

2) True

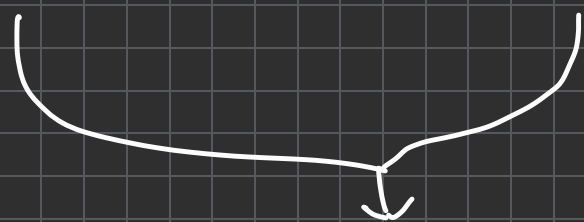
step-3: to prove your claim

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

Find two positive constants c and n_0

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

And $c > 0$ and $n_0 \geq 1$



all real numbers in range
(rational and irrational)

3) $3n^2 \leq c n^3$

$c = 1$
 $n_0 =$

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

it gives us,

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

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

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

$$3 \leq 1 \text{ (False)}$$

again, $n = 2$

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

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

$$12 \leq 8 \text{ (False)}$$

again, $n=3$

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

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

$$\Rightarrow 27 \leq 27 \text{ (True)}$$

again,

$$n=4$$

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

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

$$48 \leq 64 \text{ (True)}$$

so, $n_0 = 3$

ଅଥବା ଏହା ଏକ value ଅଟେ

[ଅର୍ଥାତ୍, n_0 ଅଟେ n ଏବଂ ଏହି value ଯାହା ଉପରେ]

statementଟି true. Not necessary ଯେ

n_0 ଯେ n (ନ ଯାହା ଏକ set ଅଟେ) ଏହା least value ଅଟେ ଅଟେ।

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

NB: c ଏହା value usually ଅଟେ ଯେ ଯେତେବେଳେ
better

Que Verify,

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

soln:

$$C = 10$$

$$n_0 = 1$$

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, $2n^2 \leq 2n^2$

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

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

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

[এখানে $2n^2 \leq 2n^2$ দিও মুক করে given equation left side এ term সূত্র add করে দেবে and \leq এ symbol কে true hold করে জন্য right side এ তার power এর term add করে দেবে।

so here's the shortcut

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

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

basically coefficient sum add krni]

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 $n = 2$,

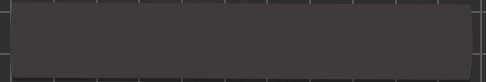
$$2(2)^2 + 3(2) + 5 \leq 10(2)^2$$

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

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

so, lets take $n = 1$.

— X —



NB: invalid statement \nexists ? proof

मिस्टर Π , just claim False

ਕਹਾਨਾ ਪ੍ਰਮਾਣਿਤ: lower bound ਤੋਂ
any program that has lower
or similar time complexity as it.

ਤਰ ਕਹਾਨਾ algorithm ਨੂੰ best case
scenario ਤੋਂ program ਨੂੰ ਉਸੇ time
complexity ਨੂੰ ਅਥਵਾ ਜਾਂ ਪ੍ਰਯੋਗ
ਨਾ ਪ੍ਰਮਾਣਿਤ: ਅਥਵਾ ਅੰਤਰ ਨੂੰ ਦਿਖਾਏ।

ਪਰੰਤੂ best case scenario-ਨੂੰ time
complexity ਤੋਂ ਪ੍ਰਮਾਣਿਤ 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,

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

$$n=1 \quad \underline{2(1)}, \quad 10 \geq 1 \text{ (True)}$$

$$n=2 \quad \underline{2(2)}, \quad 19 \geq 1 \text{ (True)}$$

so,

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