Date

ENVESTNET YODLEE

Space and Time complexity

Time complexity ! = Time taken

- Function that gives the relation about how time will grow as the input grows.

- Always look for worst case.

- Always look for complexity for larger data.

- Big-oh (O) (worst case)

- upper bound

A mathematical notation that describe the limiting behavior of a fun^n when argument tends toward a particular value.

Date

Mathematically:

$$f(N) = O(g(N))$$

$$\boxed{\lim_{N \to \infty} \frac{f(N)}{g(N)} < \infty}$$

Example:

$$f(N) = 6N^3 + 3N + 5$$

$$O(N^3) = g(N)$$

$$\lim_{N \to \infty} \frac{6n^3 + 3n + 5}{n^3}$$

$$\lim_{N \to \infty} 6 + \frac{3}{n^2} + \frac{5}{n^3}$$

substituting the value

$$6 + \frac{3}{\infty} + \frac{5}{\infty} = 6$$

$$\boxed{6 < \infty} \to \text{finite value}$$

- Big - omega (Ω) (Best case)

Lower bound

- It will be never ~~#~~ less than lower bound.

Mathematicaly :

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

Question : what if an algo has LB & UB as (N^2)

$= O(N^2)$ & $\Omega(N^2)$

$= \Theta(N^2)$

- Theta Notation (Θ)

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

Date

ENVESTNET YODLEE

- little o notation:

loose upper bound

Big oh	little oh
$f = O(g)$	$f = o(g)$
• Growth of f is no faster than g. i.e $\boxed{f \le g}$	Strictly slower than g $\boxed{f < g}$

Example

$f(n) = n^2$, $g(n) = n^3$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{n^2}{n^3} = \frac{1}{n} = 0$$

Date ________

ENVESTNET YODLEE

- little ~~ω~~ Ω notation

Loosely lower bound

· Big Ω	little ω
$f = \Omega(g)$	$f = \omega(g)$
$f \geq g$	$f > g$

Example

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

- space complexity

  - Input space + auxiliary space

→ auxiliary space (~~sp~~ extra space taken by algorithm)

Date

Question

```
for (i=1 ; i<=N ;)
{
    for(j=1 ; j<=K ; j++) O(Kt)
    {
        // operation t times
    }
}
i = i+K;
```

O(Kt * no. of times outer loop is running)

outer loop:

$i = 1, 1+K, 1+2K, + \dots 1+xK$

$1+xK \leq N$

$xK \leq N-1$

$$\boxed{x = \frac{N-1}{K}}$$

times no. of loop is running

$O\left(\cancel{k}t * \frac{(N-1)}{\cancel{k}}\right)$

$O(t * (N-1))$

$O(Nt)$ lets suppose $t = 1$

$O(N)$

Time complexity for recursion:

① Linear	② Divide & Conquern
Fibonacci	~~Divide~~ Binary Search
$F(N) = F(N-1) + F(N-2)$	$F(N) = f\left(\frac{N}{2}\right) + O(1)$

Date

→ Divide & Conquer Recurrence:

How to identify?

Form

: $T(n) = a_1 T(b_1 n + \varepsilon(n)) + a_2 T(b_2 n + \varepsilon(n)) + \dots + a_K T(b_K n + \varepsilon_K(n)) + g(n)$

for $n \geq n_0$

Akra - Bazzi theorem

$$T(n) = \theta\left(n^P + n^P \int_1^n \frac{g(n)\,dn}{n^{P+1}}\right)$$

P → P such that

$a_1 b_1^P + a_2 b_2^P + \dots = 1$

$$\sum_{i=1}^{K} a_i b_i^P = 1 \quad \text{——— (1)}$$

Ex:-1

Binary Search

$$T\left(\frac{N}{2}\right) + C$$

Example:

Merge sort

$$T(N) = 2T\left(\frac{N}{2}\right) + (N-1)$$

$a_1 = 2$, $b_1 = \frac{\cancel{N}1}{2}$, $g(n) = N-1$

$K = 1$

by using 1

$$2 * \left(\frac{\cancel{N}1}{2}\right)^P = 1$$

$\cancel{2*}$ $\therefore P = 1$

Put P in formula

$$\theta\left(n^1 + n^1 \int_1^n \frac{(u-1)}{u^{1+1}}\, dn\right.$$

$$\theta\left(n + n \int \frac{1}{u} - \frac{1}{u^2}\, dn\right)$$

$$\theta\left(n + n \left[\int_1^n \frac{du}{n} - \int_1^n \frac{du}{u^2}\right]\right)$$

$$\theta\left(n + n \left[\log u - \frac{1}{u}\right]_1^n\right]$$

$$\theta\left(n + n \left[\log n + \frac{1}{n} - 1\right]\right.$$

$$\theta\left(\cancel{n} + n \log n + 1 - \cancel{x}\right]$$

$\theta(n \log n + \cancel{1}) \to$ ignored

$\theta(n \log n)$ // Time complexity

Date

For array of size N :

Merge sort complexity = O(N log n)

Question :

$$T(N) = \underset{a_1}{2}\,T\left(\frac{N}{\underset{b_1}{2}}\right) + \underset{a_2}{\frac{8}{9}}\,T\left(\underset{b_2}{\frac{3N}{4}}\right) + \underset{g(n)}{N^2}$$

$$2 \times \left(\frac{1}{2}\right)^p + \frac{8}{9}\left(\frac{3}{4}\right)^p = 1$$

let assume $P = 2$.

$$T(n) = \theta\left(n^2 + n^2 \int_1^n \frac{u^2}{n^{2+1}}\,du\right]$$

$$= \theta\left(n^2 + n^2 \int_1^n \frac{u^2}{u^3}\,du\right.$$

Date

$$\theta\left(n^2 + n^2 \int_1^n \frac{u^2}{u^3}\, du\right)$$

$$\theta\left(\cancel{n^2} + n^2 \log n\right)$$

$$\theta\left(n^2 \log n\right)$$

→ If u can't find value of P

$$T(n) = 3T\left(\frac{n}{3}\right) + 4T\left(\frac{n}{4}\right) + n^2$$

Let's try $P = 1$

$$3 \times \left(\frac{1}{3}\right)^P + 4 \times \left(\frac{1}{4}\right)^P = 1$$

~~If~~ If $P = 1$ we are getting

$2 = 1$ increase denominator

$P > 1$

Date

ENVESTNET YODLEE

$P = 2$

$$3 \times \frac{1}{9} + 4 \times \frac{1}{16}$$

$$= \frac{1}{3} + \frac{1}{4} = \frac{4+3}{12} = \frac{7}{12} < 1$$

$1 > P < 2$

Note: when $P <$ Power of $g(n)$
then ans $= g(n)$

Hence $g(n) = n^2$
$P < 2$ (i.e Power of $g(n)$)

ans $= O(g(x))$

$$T(n) = \Theta\left(n^P + n^P \int_1^n \frac{u^2}{u^{P+1}}\, dn\right.$$

$$= \Theta\left(n^P + n^P \int_1^n u^{1-P}\, du\right.$$

Date

$\theta(n^p + n^2)$
↳ ignore

Ans $O(n^2)$

→ Solving Linear Recurrence

Example:
$F(N) = F(N-1) + F(N-2)$

$f(n) = a_1 f(n-1) + a_2 f(n-2) + a_3 f(n-3) + \ldots + a_n f(n-n)$

$f(n) = \sum_{i=1}^{n} a_i f(n-i)$

~~f(n) =~~ $\sum_{i=1}^{n} a_i$

for a_i, n is fixed
n = order of recurence