Time Complexity

- Actual time taken for an algorithm doesnt matter. The function that tells how the time taken changes with respect to input size matters. It is called Time Complexity.

- Always look complexity for large input data (worst case).

- Since we care only about how the time taken changes with input size so,

  $an+b \longrightarrow$ linear.

  $n \longrightarrow$ Linear.

  So we ignore small power terms and constants. Also for higher inputs low power terms would be neglected.

Big-O Notation

- This is the upper bound. The problem will never exceed this complexity.

$f(N) \Rightarrow O(g(N))$

$$\Rightarrow \underbrace{\lim_{N \to \infty} \frac{f(N)}{g(N)}}_{\text{finite number.}} < \infty$$

$f(N)$ grows faster than $g(N)$.
Not strict
$f(N) \leq g(N)$

eg:- $f(N) = 6N^3 + 3N + 5$

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

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

$$= 6 < \infty$$

$\therefore O(N^3)$ //

Big Omega Notation

$f(N) \Rightarrow \Omega(g(N))$

$$\Rightarrow \lim_{N \to \infty} \frac{f(N)}{g(N)} > 0$$

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

Theta Notation

lower

Theta (middle)

upper

Little oh Notation

- Also gives upper bound
- ~~Not~~ strictly $f(N) < g(N)$

$$\lim_{n \to \infty} \frac{f(N)}{g(N)} = 0.$$

Little Omega

- Also gives lower bound.
- $$\lim_{n \to \infty} \frac{f(N)}{g(N)} = \infty.$$

Space Complexity.

we consider.

- Input space + Auxilliary Space

Recursive Algorithms.

- Only calls that are interlinked will be present on stack at same time.

In recursion tree, no funct^n call of same level will be present at same time.

- So space complexity will be the height of tree. for recursive algorithm

Recurrence Relations

Linear

↓

eg:- $f(n) = f(n-1) + f(n-2)$

Divide & Conquer

↓

eg:- $f(n) = f(N/2) + O(1)$

Divide & conquer recurrences

- Form. (For identification)

$$T(x) = a_1 T(b_1 x + \varepsilon_1(x)) + a_2 T(b_2 x + \varepsilon_2(x)) + \dots + a_k T(b_k x + \varepsilon_k(x)) + g(x)$$

for $x \geq x_0 \rightarrow$ const.

eg:- Take B.S.

$$T(N) = T(N/2) + c;$$

$a_1 = 1$ $\quad g(x) = c$

$b_1 = 1/2$

$\varepsilon_1(x) = 0$

meaning of $g(n)$

- After returning what we does with it.

* for BS
$$f(n) = f(n/2) + c \rightarrow g(n).$$
constant for the conditionals (checks)

* for merge sort,
$$f(n) = f(n/2) + f(n/2) + n - 1 \rightarrow g(n)$$
no. of comparisons

Solving Divide & Conquer Recurrence

Akra Bazzi formula

$$T(x) = \Theta\left(x^p + x^p \int_1^x \frac{g(u)}{u^{p+1}}\, du\right)$$
in simplest form.

- finding p,
$$\sum_{i=1}^{k} a_i b_i^p = 1$$
and solve $T(x)$

- Not able to find p,

$$T(n) = 3T(n/3) + 4T(n/4) + n^2$$

$$\begin{array}{ccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ a_1 & b_1 & a_2 & b_2 & g(n) \end{array}$$

$$\sum_{i=1}^{k} a_i b_i^p = 3 \times \frac{1}{3}^p + 4 \times \frac{1}{4}^p$$

let $p = 1$, $3 \times 1/3 + 4 \times 1/4$

$= 2$, this should be 1

means 2 should reduce to 1 or take bigger p.

let $p = 2$,

$$3 \times \frac{1}{3}^2 + 4 \times \frac{1}{4}^2$$

$$= \frac{7}{12} < 1$$

So, $1 < p < 2$.

If $p <$ power of $g(n)$,

Complexity is $O(g(n))$

Solving Linear Recurrences

Form.

Homogenous Linear.

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

$n \rightarrow$ order of recurrence

Solution.

Take fibo(n)

$f(n) = f(n-1) + f(n-2)$ — ①

step - 1 :

put $f(n) = \alpha^n$ for some α.

$$\alpha^n = \alpha^{n-1} + \alpha^{n-2}$$

$$\alpha^n - \alpha^{n-1} - \alpha^{n-2} = 0$$, divide by α^{n-2}.

$$\alpha^2 - \alpha - 1 = 0$$

Solve this,

$$\alpha = \frac{1 \pm \sqrt{5}}{2}.$$

$$\alpha_1 = \frac{1+\sqrt{5}}{2} \qquad \alpha_2 = \frac{1-\sqrt{5}}{2}.$$

Step 2

$f(n) = c_1 \alpha_1^n + c_2 \alpha_2^n$ is a sol^n of $f(n)$.

$$f(n) = c_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + c_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$$

Fact $\Rightarrow$ Here we have two roots α_1 & α_2 so we should have two answer $f(0) = 0$ & $f(1) = 1$

$$\therefore \quad c_1 \left(\frac{1+\sqrt{5}}{2}\right) + c_2 \left(\frac{1-\sqrt{5}}{2}\right) = 1 \quad \Rightarrow f(1) = 1$$

$$c_1 + c_2 = 0$$

$$c_1 = -c_2 \longrightarrow f(0) = 0$$

$$\therefore \quad c_1 \left(\frac{1+\sqrt{5}}{2}\right) - c_1 \left(\frac{1-\sqrt{5}}{2}\right) = 1$$

$$c_1 = \frac{1}{\sqrt{5}}, \quad c_2 = -\frac{1}{\sqrt{5}}$$

$$\therefore f(n) = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{-1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n$$

it is actually formula for n^{th} fibonacci number.

Time Complexity:-

$$f(n) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

$\hookrightarrow < 1$ so for big n it can be neglected

so $f(n) = O\left(\frac{1+\sqrt{5}}{2}\right)^n$

$= O\{(1.618)^n\}$.

Equal roots case

$$f(n) = 2f(n-1) - f(n-2)$$

$f(n) = \alpha^n$

$\alpha^n = 2\alpha^{n-1} - \alpha^{n-2}$

$\Rightarrow \alpha^2 - 2\alpha + 1 = 0$

$\Rightarrow \alpha = 1 \rightarrow$ double root

- If α is repeated r times,

$\alpha^n, n\alpha^n, n^2\alpha^n \ldots n^{r-1}\alpha^n$ are

all solutions of $f(n)$

for above qst 2 identical roots

so roots would be α^n and $n\alpha^n$.

$$\therefore f(n) = c_1(\alpha)^n + c_2(n\alpha^n)$$

$$\alpha \Rightarrow 1$$

$$f(n) = c_1 + c_2 n$$

$f(0) = 0$
$f(1) = 1$

$$f(0) = 0 = c_1$$

$$f(1) = c_1 + c_2 \times 1 = 1$$

$$c_2 = 1$$

$$\therefore f(n) = n \rightarrow O(n)$$

Non homogeneous Linear recurrence

<u>form</u>

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \ldots + a_d f(n-d)$$

$$+ g(n)$$

$\downarrow$

extra function $\rightarrow$ non homogeneous step-1

replace $g(n)$ by 0 and solve homogeneously

eg:- $f(n) = 4f(n-1) + 3^n$, $f(1) = 1$

$$\Rightarrow f(n) = 4f(n-1).$$

$$f(n) = \alpha^n$$

$$\alpha^n = 4\alpha^{n-1}$$

$$\Rightarrow \alpha - 4 = 0$$

$$\alpha = 4.$$

$$f(n) = c_1 \alpha^n$$

$$\cancel{f(1) = 1 = c_1 4}$$

$$\cancel{c_1 = 1/4}$$

Step-2

put $g(n)$ on one side and find particular solution.

$$f(n) - 4f(n-1) = 3^n.$$

for $f(n)$ put something identical to 3^n say $c3^n$,

$$c3^n - 4c3^{n-1} = 3^n$$

$3c - 4c = 3$

$c = -3$

$\therefore c\,3^n = -3 \times 3^n = -3^{n+1}$

Add homogeneous and particular for final solutⁿ,

~~$f(n) = 1/4\ 4^n$~~

$f(0) = c_1 4 - 3^2 = 1$

$c_1 = 5/2$

$f(n) = \frac{5}{2} 4^n - 3^{n+1}$

— How to guess for particular solution

- If $g(n)$ is exponential,

$g(n) = 2^n + 3^n$

we can take $a2^n + b3^n$

- for polynomials,

$g(n) = n^2 - 1$

$\rightarrow an^2 + bn + c$.

- If for 2^n, $a2^n$ fails, then take $(an+b)2^n$. If it fails too, take $(an^2+bn+c)2^n$.