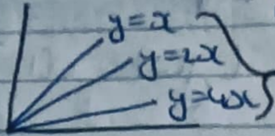
TIME AND SPACE COMPLEXITY

- time taken $\boxed{\times}$, function showing how time grows wrt input $\boxed{\checkmark}$
- worst case scenario $\boxed{\checkmark}$ $(n \to \infty)$
- 

even though linear, value is diff (! same time taken)
but we don't consider time taken
as it depends on machine $\therefore$ $\boxed{\text{IGNORE CONSTANTS}}$

same time complexity of $O(n)$

- ignore less dominating terms $\boxed{\checkmark}$
- $O(1) < O(\log N) < O(N) < O(N \log N) < O(2^N) < O(N!)$

BIG OH $\Rightarrow$ (upper bound) $(O())$

$f(n) \Rightarrow O(g(n))$
$\Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$ $\boxed{f \leq g}$

BIG OMEGA $\Rightarrow$ (lower bound) $(\Omega())$

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

BIG THETA $\Rightarrow$ (combined) $(\Theta())$

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

NOTE $\Rightarrow$ Little oh $\Rightarrow$ $f < g$ $(\lim = 0)$, Little omega $\Rightarrow$ $f > g$ $(\lim = \infty)$

Space complexity $\Rightarrow$ Input space + Auxiliary space
(extra space taken by the algorithm)

example $\Rightarrow$ in binary search,
aux space $\Rightarrow$ constant (3 vars) always, irrespective of input size
and,
space complexity $\Rightarrow O(n)$

Remarks ⇒ form for divide and conquer recurrence relation ⇒

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

for $x \geq c$

eg ⇒ for binary search ⇒ $T(N) = T(N/2) + O(1)$

$(a_1 = 1, b_1 = 1/2, \varepsilon_1(x) = 0)$ and $(g(x) = O(1))$

AKRA BAZZI THEOREM (for divide and conquer)

$$T(x) = \Theta\left(x^p + x^p \int_1^x \frac{g(u)}{u^{p+1}}\, du\right) ; \sum_{i=1}^{k} a_i b_i^p = 1$$

example ⇒ $T(x) = 3T(x/3) + 4T(x/4) + x^2$

for finding p, let's try $p = 1$,

$3 \times \left(\frac{1}{3}\right)^1 + 4 \times \left(\frac{1}{4}\right)^1$ ⇒ 2 which is > 1

∴ let's try $p = 2$ ⇒ $7/12 < 1$ it means. $p < 2, > 1$.

But if $p <$ power of $g(x)$ (here, 2) then ans $= g(x)$

∴ here, as $p < 2$ ⇒ ans $= O(g(x)) = O(x^2)$

Remarks ⇒ form for linear recurrences ⇒

$$f(x) = \sum_{i=1}^{n} a_i f(x-i) \quad ; \; n = \text{order of recurrence}$$

SOLVING LINEAR RECURRENCES

example ⇒ consider fibo seq. ⇒ $f(n) = f(n-1) + f(n-2)$ ①

characteristic eqn for recurrence

STEP 1 – put $f(n) = \alpha^n$ for some constant α

$$\Rightarrow \alpha^n = \alpha^{n-1} + \alpha^{n-2} \Rightarrow \alpha^n - \alpha^{n-1} - \alpha^{n-2} = 0$$

$\Rightarrow$ on solving, $\boxed{\alpha = \frac{1 \pm \sqrt{5}}{2}}$; (α_1 = '+' one and α_2 = '−' one)

STEP 2 – put $f(n) = c_1\alpha_1^n + c_2\alpha_2^n$

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

STEP 3 – no. of roots = no. of ans you have already

$\Rightarrow$ here as α count = 2 means we have 2 ans already which are $F(0)=0$ and $F(1)=1$.

Now, $F(0) = 0 = c_1 + c_2$ (ii; $n=0$)

$$\Rightarrow c_1 = -c_2 \quad \text{— (iii)}$$

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

Putting (iii) above, $\boxed{c_1 = 1/\sqrt{5},\ c_2 = -1/\sqrt{5}}$ — (iv)

Putting (iv) in (ii) $\Rightarrow$

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

FORMULA FOR N^{th} FIBO NUMBER

Now for the time complexity $\Rightarrow \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$

ignore this becoz as $n\uparrow$ it becomes close to 0 $\therefore$ less dominating.

Now after ignoring constants also, time complexity $\Rightarrow$

$$\boxed{O\left(\frac{1+\sqrt{5}}{2}\right)^N}$$

NOTE ⇒ In general scenarios if roots are repeated r times say then we can take any one from below as ~~the second root~~ other roots
⇒ $\alpha^n, n\alpha^n, n^2\alpha^n \ldots n^{r-1}\alpha^n$

For eg ⇒ $a = 1, 1$ then $\alpha_1 = 1$ and we can take α_2 as $n\alpha^n \Rightarrow \alpha_2 = n$.

- SOLVING NON HOMOGENOUS LINEAR RECURRENCES

FORM ⇒ $\boxed{f(x) = \sum_{i=1}^{k} a_i f(x-i) + g(x)}$

STEP 1 – Replace $g(x)$ with 0 and solve as usual. let's say we got $\alpha = 4$, ⇒ $f(x) = c_1 4^x$

homo. sol.

STEP 2 – take $g(x)$ on one side and find particular solution
let's say $g(x) = 3^x$

⇒ $f(x) - 4f(x-1) = 3^x$

STEP 3 – for $g(x) = x^n$ guess a polynomial of degree n ~~and replace f(x) with it~~.

let's say guess $= c3^x$ ⇒ $f(x) = c3^x$ —(I)

∴ $c3^x - 4c3^{x-1} = 3^x \Rightarrow c = -3$ —(II)

putting (II) in (I) ⇒ $\boxed{f(x) = -3^{x+1}}$

(particular solution)

STEP 4 – add both solⁿ together.

$\Rightarrow f(n) = c_1 4^n - 3^{n+1}$

Now proceed as usual like if the solⁿ we know is

$f(1) = 1 \Rightarrow c_1 = 5/2 \Rightarrow \boxed{f(n) = \frac{5}{2} 4^n - 3^{n+1}}$

✡ how to ~~get a p~~ guess?

→ If $g(x)$ = exponential guess of same type
eg ⇒ $g(x) = 2^n + 3^n \Rightarrow$ guess $= a2^n + b3^n$.

→ If polynomial, guess of same degree.
eg ⇒ $g(x) = n^2 - 1 \Rightarrow$ guess $= an^2 + bn + c$

→ If combined like $g(x) = 2^n + n$
then $f(n) = a2^n + (bn + c)$

✡ let's say you guessed $f(x) = a2^n$ and it fails, then try $(an+b)2^n$ and if this also fails then try $(a^2n + bn + c)2^n$ ....