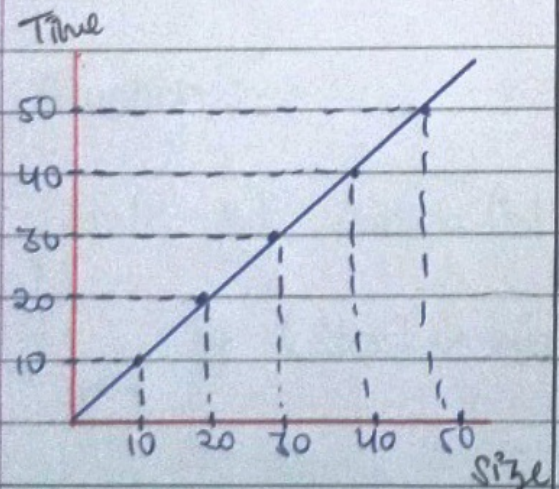
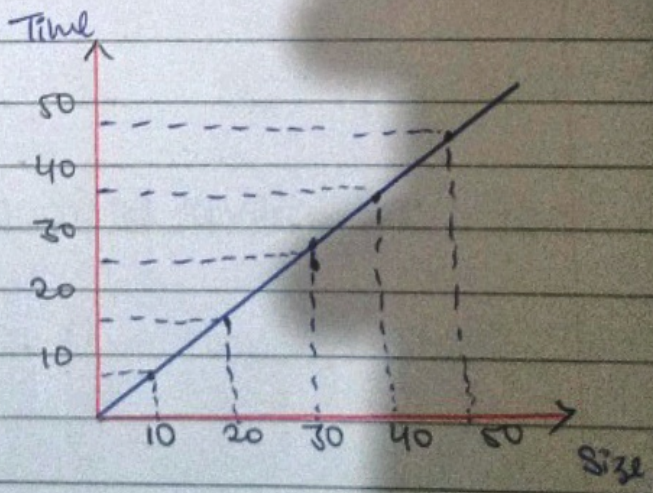
• Time and Space Complexity :-

– Function that tells how the time is going to grow as the size is growing. It is known as time complexity.

Time Complexity != time taken

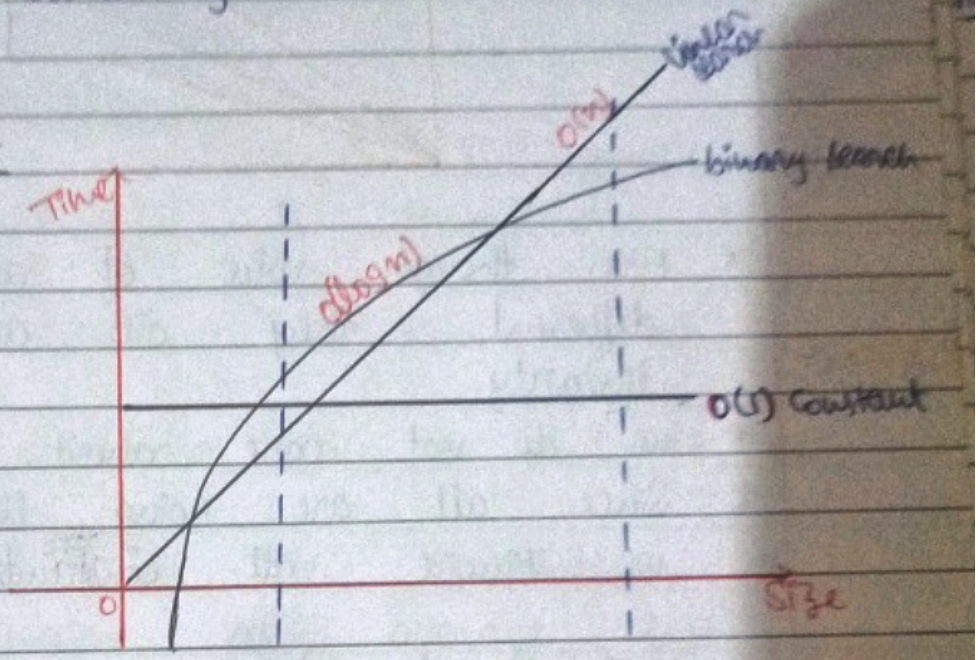
	Old machine	M1 macbook
data →	10,000 Elements	data → 10,000 Elements
algo →	linear search	algo → linear search
TT →	10s	TT → 1s





- both machines have same time complexity. which is of linear time complexity.
  " we can see in the graph that when the size is increasing the time is also increasing. "

- **Importance :-**



$\Rightarrow \{O(1) < O(\log N) < O(N)\}$

as we can see the graph that for small data size linear is taking less time but we do not care about small data size.

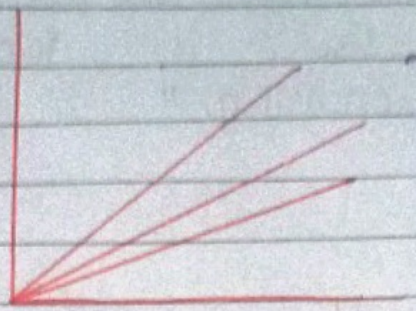
Ques ⇒ what do we consider when thinking about complexity ?

⇒ Always look for worst case complexity.

Date: / / Page no:

ii) always look at complexity for large ∞ data.

iii)



* Even though value of actual time is different, they all are growing linearly.
* we do not care about actual time.
* Since all are going linear that's why we ignore all constants.

iv) $O(N^3 + \log N)$

* from point 2 — always ignore less dominating terms.

why? —→

Ex ⇒ 1 mln data = $O((1\text{mln})^3 + \log(1\text{mln}))$

$= (1\text{mln})^3 + 6$

So we can ignore 6. Since it is very small.

Ex:- $O(N^3 + 4N^2 + 5N + 6)$

⇒ $O(N^3 + N^2 + N)$ (point 3)

⇒ $O(N^3)$ (point 4)

- **Big O notation :-**

It is upper bound means it will never exceed this time limit. That's why it is known as worst case complexity. ⟶ $O(N^3)$

- **Big omega notation :-**

It is opposite of big O. Means it is presentation of lower bound.

$O(N^3)$ → min time required.

ques⇒ what if algo has a ub and lb as N^2 ?

ans→ In that case we use θ notation. It is also known as average case. It combines big O and big omega.

- **Space Complexity :-**

Space Complexity is combination of input space and auxiliary space.

| Space Complex = input + auxiliary space |

auxiliary space is the extra space used by the algorithm.

Date: / / Page no:

we always talk about auxiliaries space because we can't control the input size.

Ans⇒ find the time complexity :-

```
for(i=1 ; i<=N)
{
    for(j=1 ; j<=k ; j++)
    {
        // Some * operation that takes + time.
    }
    i= i+k;
}
```

ans⇒ inner loop :- O(kt) time

outer loop :-

$i = 1, 1+k, 1+2k + \dots + 1+xk$

so,

$1+xk \leq N$

$xk \leq N-1$

No of times outer loop running ← $x = N-1/k$

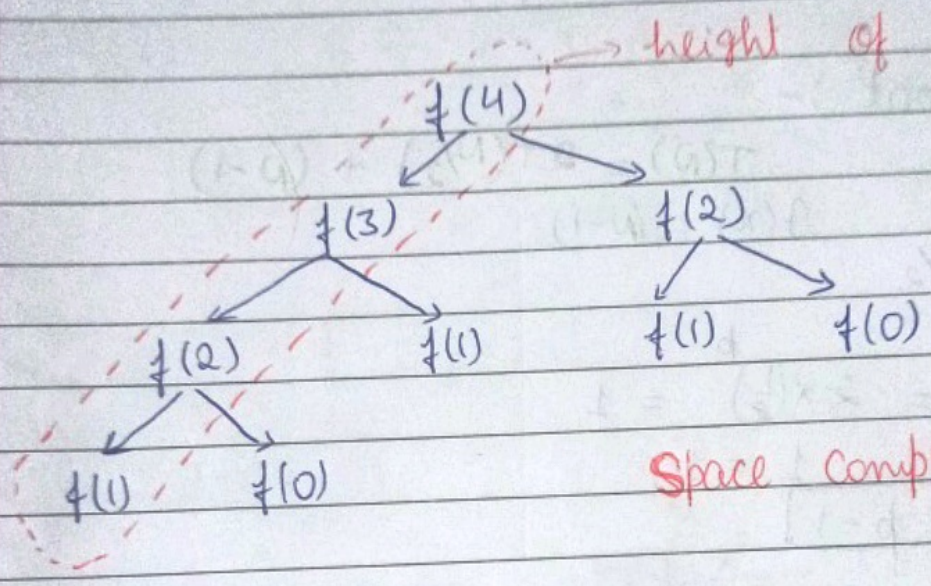
total time ⇒ $O(kt * \frac{(N-1)}{k})$

$\Rightarrow O(Nt - t)$

$\Rightarrow O(N)$ ans.

- **Recursive algorithm :-**

Space complexity of recursive program = height of tree.



Space complexity $\Rightarrow O(N)$

- 2 types of recursion :-
  - i) linear
  - ii) divide and conquer.

Ques ⇒ How to solve to get complexity :-

- i) plug and chug (ignore)
- ii) Master's theorem (ignore)
- iii) Akra bazzi formula (✓)

Date: / / Page no:

- **Akrabazzi formula :-** for divide & conquer.

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

$$\left\{ p :- \sum_{i=1}^{k} a_i b_i^p = 1 \right\}$$

Ex:- Merge sort :-

$$T(N) = 2T(N/2) + (N-1)$$

$a_1 = 2$ $\quad g(x) = (N-1)$

$b_1 = 1/2$

$$p = 2 \times \left(\frac{1}{2}\right)^p = 1$$

$$1 = 1$$

$\therefore p = 1$

apply formula :-

$$T(N) = \Theta\left(n^1 + n^1 \cdot \int_{x_1}^{n} \frac{(N-1)}{N^2}\, dn\right)$$

$$= \Theta\left(n + n \int_1 \frac{1}{N} - \frac{1}{N^2}\, dn\right.$$

$$= \Theta\left(n + n \left[\log N - \frac{1}{N}\right]_1^N\right.$$

$$= \Theta\left(n + n \log N - n/n - n\right)$$

$$= \Theta(n \log n) \quad \text{ans.}$$

Ans → $T(N) = 2T(\frac{N}{2}) + 8/9\, T(\frac{3N}{4}) + N^2$

$a_1 = 2T$ $\quad a_2 = 8/9$ $\quad g(U) = N^2$

$b_1 = N/2$ $\quad b_2 = 3/4$

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

Trial and error ($p=2$)

$$p = 2 \times \frac{1}{4_2} + \frac{8}{8} \times \frac{9}{16_2}$$

$$= \frac{1}{2} + \frac{1}{2}$$

$$\Rightarrow 1$$

$$\therefore p = 2$$

formula :-

$$T(N) = \theta\left(x^2 + x^2 \cdot \int_1^x \frac{x^2}{x^3}\, dx\right)$$

$$= \theta\left(x^2 + x^2 \cdot \int_1^x \frac{1}{x}\, dx\right)$$

$$= \theta\left(x^2 + x^2 \cdot [\log x]_1^x\right)$$

$$= \theta\left(x^2 + x^2 \log x\right)$$

$$\text{ans} = \theta(x^2 \log x)$$

- what if we can't find the value of p?

→ → →

Date: / / Page no:

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

$a_1 = 3$ $\quad b_2 = 1/4$

$b_1 = 1/3$ $\quad a_2 = 4$

try $p=1$

$$3 \times \frac{1}{3} + 4 \times \frac{1}{4} = 1$$

$$2 \neq 1$$

$\therefore p > 1$

try $p=2$

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

$$\frac{1}{3} + \frac{1}{4} = 1$$

$$\frac{7}{12} \neq 1$$

$\therefore p < 2$

Note :- when $p <$ power of $(g(n))$ then ans is $g(n)$.

here $g(n) = n^2$

ans $= O(n^2)$

* For linear recurrance :-

$$F(n) = F(n-1) + F(n-2) \quad \text{(fibonacci)}$$

forum :- $f(n) = a_1 f(n-1) + a_2 f(n-2) + \cdots + a_n f(n-n)$

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

Solution for fibonacci :-

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

Steps :-

i) put $f(x) = \alpha^x$ for constant α

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

divide by $\alpha^{(n-2)}$

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

$$\Rightarrow \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

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

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

Date: / / Page no:

ii) $f(n) = c_1\alpha_1^n + c_2\alpha_2^n$ is also a solution for fibonacci.

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

iii) fact :- No of root = No of eqns you have already.

$$f(0) = 0 = c_1 + c_2$$
$$\Rightarrow c_1 = -c_2 \longrightarrow (3)$$

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

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

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

putting this in eqn (2) -

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

formula for nth fibonacci no.

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

→ as the n increases it will head towards 0. So ignore the less dominating term.

Time complexity :-

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

↓

Golden Ratio

- Non - homogeneous linear recurrance relation :-

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \cdots + a_d f(n-d) + g(x)$$

↓

This extra function is known as non-homogeneous.

How to solve?

i) Replace $g(x)$ by 0 and solve usually.

Ex:- $f(n) = 4f(n-1) + (3^n) \rightarrow 0$

$= 4f(n-1)$

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

$\alpha^n - 4\alpha^{n-1} = 0$

$\boxed{\alpha = 4}$

Date: / / Page no:

homogeneous solution :-

$$f(n) = C_1 2^n$$
$$\boxed{f(n) = C_1 4^n}$$

ii) Take g(n) on one side and find particular solution :-

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

Guess something similar to g(n)

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

guess :- $f(n) = C3^n$

$$C3^n - 4C3^{n-1} = 3^n$$

$$\boxed{C = -3}$$

particular solution :- $f(n) = C3^n$

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

iii) Add both solution together :-

$$f(n) = C_1 4^n - 3^{n+1}$$

$$f(1) = 1 \Rightarrow C_1 4 - 3^2 = 1$$

$$C_1 = 5/2$$

$$\boxed{f(n) = 5/2\ 4^n - 3^{n+1}}$$ ans.

* How do we guess a particular solution —

i) if $g(n)$ is exponential guess the same type.

Ex:- $g(n) = 2^n + 3^n$

→ guess:- $f(n) = a2^n + b3^n$

ii) if $g(n)$ is polynomial, guess of same degree.

$g(n) = n^2 - 1$ ⟶ degree 2

→ $f(n) = an^2 + bn + c$

iii) for combination of both -

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

guess:- $f(n) = a2^n + (bn + c)$