Factor of a number n

↳ it a number that completely divides n.

for eg: x is a factor of n.

then n % x = 0

8 → 1, 2, 4, 8

Quiz 1 : 24 → 1, 2, 3, 4, 6, 8, 12, 24.

24

1 x 24

2 x 12

3 x 8

4 x 6

6 x 4

8 x 3

12 x 2

24 x 1

Quiz 2 $\quad 10 \rightarrow [1 \quad 2 \quad 5 \quad 10]$

Q1 Given a number n, find out the total number of factors.

BRUTE FORCE

```
int count = 0
for (int i = 1 ; i <= n ; i++) {
    if (n % i == 0) {
        count ++ ;
    }
}
```

3

3

Iteration : 1 loop.

The above code has n iterations

10^8 iterations = 1 sec.

n	no of iteration	Time taken
10^8	10^8	1 sec
10^{10}	10^{10}	100 sec
10^{18}	10^{18}	10^{10} seconds

↳ 317 years!

10^8 iteration = 1 sec

1 iteration = $\frac{1}{10^8}$ sec.

10^{10} iteration = $\frac{1}{10^8} \times 10^{10}$

$= \frac{10^{10}}{10^8} = 10^{10-8}$

$= 10^2$

$\frac{x^a}{x^b} = x^{a-b}$

n : i is a factor

then $\frac{n}{i}$ is also a factor.

$j = \frac{n}{i}$

24.

i	n/i
1	24
2	12
3	8
4	6
6	4
8	3
12	2
24	1

$i \leq \frac{n}{i}$

$\Downarrow$

$i \times i \leq n$

$i \leq \sqrt{n}$

25

i	n/i
1	25
5	5
25	1

```
for (int i = 1 ; i*i <= n ; i++) {
    if (n % i == 0) {
        if (i != n/i)
            count = count + 2 ;
        else
            count = count + 1 ;
```

3

3

24

i	count
1	2
2	4
3	6
4	8

25

i	count
1	2
5	3

Total iterations : $\sqrt{n}$.

n	Total iteration.	Total time
10^{18}	10^9	10 seconds

Root of number

$$\sqrt{n} = a$$

$$a \times a = n.$$

$$\sqrt{36} = 6$$

$$6 \times 6 = 36.$$

$$\sqrt{a^b} = a^{b/2}$$

$$\sqrt{10^{18}} \Rightarrow 10^{18/2} = 10^9$$

Prime numbers : Number where
count of factors = 2

~~10~~ 11 23 2 ~~25~~ ~~27~~ 31

= 4

[a, b] ⇒ Range of numbers where both a & b are inclusive.

(a, b) ⇒ Range of numbers where both a & b are exclusive.

Quiz 4 [3, 10]

10 - 3 + 1 = 8

3, 4, 5, 6, 7, 8, 9, 10

[a, b]

[b - a + 1]

⇓

no of integers in the range [a, b]

Sum of 1st n natural number $= \boxed{\frac{n(n+1)}{2}}$

Natural number : 1, 2, 3, 4 . . . .

$1 + 2 + 3 + 4 \quad \ldots \ldots \quad 100$

$\Rightarrow \frac{100(101)}{2} = 5050$

Geometric Progression [G.P]

$\Rightarrow$ 5, 10, 20, 40, 80, 160

$\Rightarrow$ 4, 12, 36, 108

$\Downarrow$

$\frac{12}{4} = \frac{36}{12}$

r = common ration

a = first term

Sum of 1st n terms in GP.

$a, ar, ar^2, ar^3 \text{ ---- } ar^{n-1}$

$$S = \frac{a(r^n - 1)}{r - 1} \Rightarrow r \neq 1.$$

5, 10, 20, 40, 80, 160

$a = 5, \quad r = 2, \quad n = 6$

$$\frac{a(r^n - 1)}{r - 1} \Rightarrow \frac{5(2^6 - 1)}{2 - 1}$$

$$\Rightarrow 5(64 - 1)$$

$5 \times 63 \Rightarrow 315$

Q for (int i = 1 → n) {

if (i == n)

break;

}

$[1, n] \Rightarrow n - 1 + 1$

$\Rightarrow n$ iterations

Q for (int i = 0 → 100) {

S = S + i + i^2 ;

}

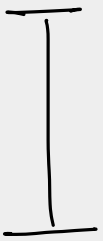
$$[0, 100] = 100 - 0 + 1$$

⇒ 101 iterations.

Quiz 2

How many iterations will be there in this loop?

```
func(){
    for(i -> 1 to N){
        if(i % 2 == 0){
            print(i);
        }
    }
    for(j -> 1 to M){
        if(j % 2 == 0){
            print(j);
        }
    }
}
```



n iterations

m iterations

$= n + m$

How to compare 2 algorithms!

Sumanth	Amogh
10 sec	20 sec
	↓ writes in C++
10 sec	8 sec
↓ write in mac	
5 sec	8 sec

Time is not the best metric to compare 2 algorithms because time gets affected by external factor.

No of iteration ⟹ Best metric to compare algorithms.

Sumanth

$100 \log n$

Amogh

$n/10$

$n \leq 3500$ | Amogh was better

$n > 3500$ | Sumanth was better.

Most watched video on youtube $\Rightarrow$ Somewhere in billions.

Ind vs Pak $\Rightarrow$ viewship in crore.

Asymptotic Analysis of Algorithms : Analysing algorithm for very large number $= \infty$

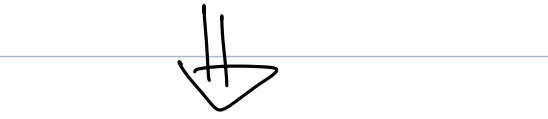
BIG O

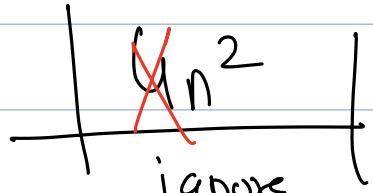
1) no of iterations.

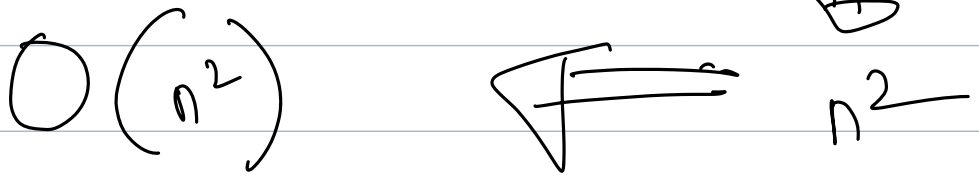
2) ignore lower order terms

3) ignore constants.

Ex 1 no of iterations : $4n^2 + 2n + 10$







Quiz $f(n) = 4n + 3n\log n + 1$

$4n$ ~~$3n\log n$~~ ~~1~~

~~3~~$n\log n$

$= O(n\log n)$

Q $F(n) = 4n\log n + 3n\sqrt{n} + 10^6$

$= O(n\sqrt{n})$

Comparision Order.

$\log(n) < \sqrt{n} < n < n\log n$
$< n\sqrt{n} < n^2 < n^3 < 2^n < n^n$

Neglect Lower Order terms.

$$\text{Iterations} = n^2 + 100n.$$

Input Size	Total iterations	Contribution of lower order term.
$n = 10$	$100 + 1000$	90%
$n = 100$	$10000 + 10000$	50%
$n = 10^5$	$10^{10} + 10^7$	$< 0.1\%$

As input size increases contribution of lower order term decreases.

Neglect Constants.

Algo 1	Algo 2	Which is better for large inputs.
$10 \log n$	n	Algo 1
$100 \log n$	n	Algo 1
$1000 \log n$	n	Algo 1

Issues with Big O

1st Issue :

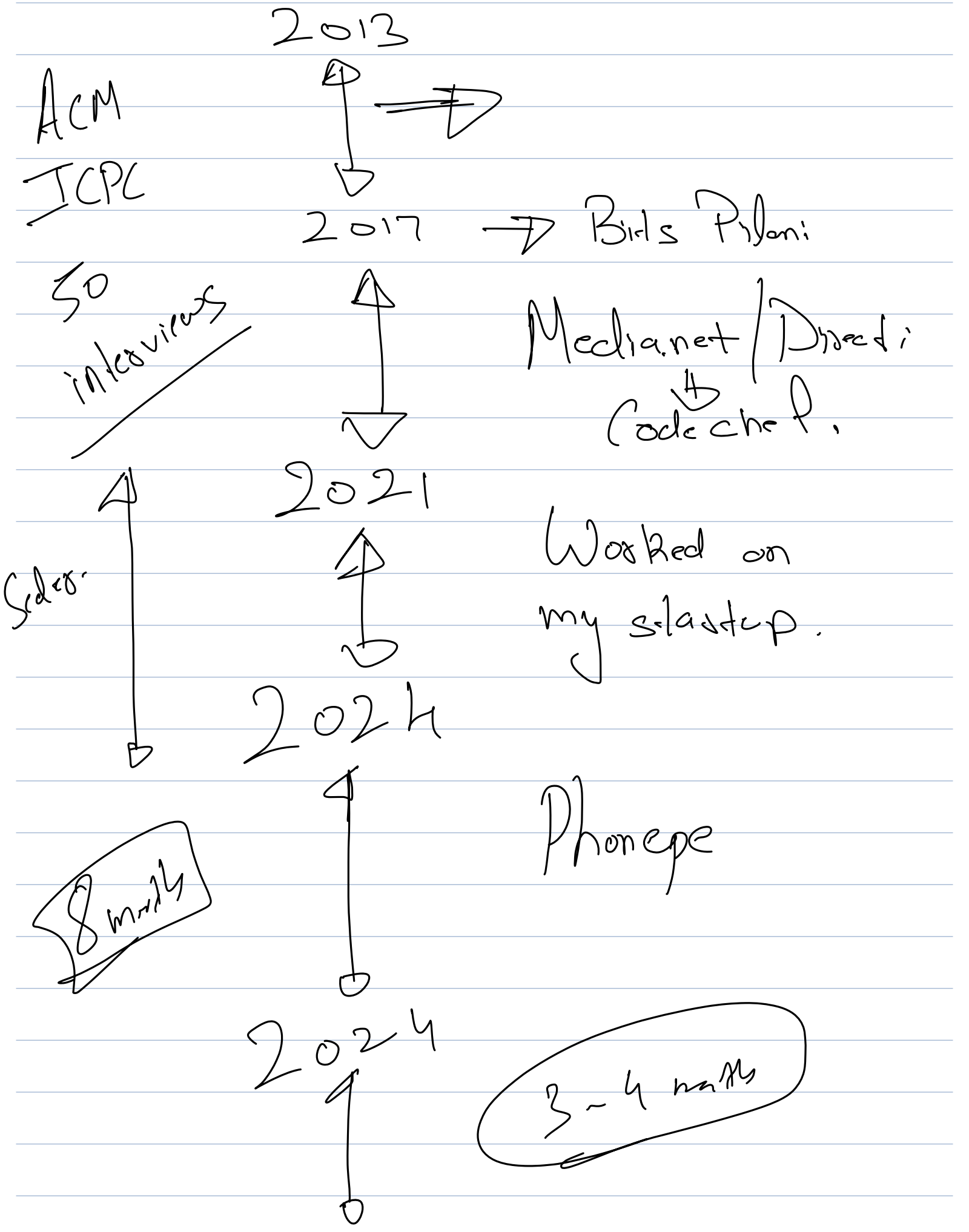
When we say that algo 1 is better than algo 2 using Big O, we can only make ~~that~~ that claim for large input.

2nd Issue.

Algo 1 : $2n^2 + 4n$ $\Rightarrow$ $O(n^2)$

Algo 2 : $3n^2$ $\Rightarrow$ $O(n^2)$

When Big O is same, we might have to look at number of iterations to say which is better.


2013
ACM ICPC
2017
Birla Pilani
Mediamet / Directi
Codechef.
50 interviews
2021
Worked on my startup.
Selected
2024
Phonepe
8 months
2024
3-4 months