#### Welcome!!!

## Few Terms that you shall see/hear throughout the course:

- 1. PSP (Problem Solving Percentage) Solved Assignment Problems / Total Open Assignment Problems
- \* There are two types of section Assignment and Additional. Assignment section consists of implementation of the problems done in class. PSP is calculated based on only Assignment Problems.
- \* Additional Problems are slight modifications of assignment problem, they are not part of PSP but once you're done with assignment, we highly recommend to complete additional problems as well.
- \* Try to keep PSP least 90% no matter what. It shall really help you to stay focused and we have seen in the past that people with >= 90%, do well in Interviews.

#### 2. Attendance

- \* Try to maintain at-least 80% attendance either through live classes or by watching recording.
- \* Though I will recommend you to come to classes regularly because otherwise it may create backlogs.
- \* So, I expect all of you to attend live classes and if for any reason you are unable to, then please send a message stating the reason in the WhatsApp group.

### **Intermediate Module Description**

- \* Introduction to Problem Solving
- \* Time Complexity
- \* Introduction to Arrays
- \* Prefix Sum
- \* Carry Forward
- \* Subarrays
- \* 2D Matrices
- \* Sorting Basics
- \* Hashing Basics
- \* Strings Basics

- \* Bit Manipulation Basics
- \* Interview Problems
- \* Contest [covers Full Intermediate DSA]

#### FAQs:

- \* Notes will be uploaded after the class.
- \* Assignments will be unlocked after the class ends.
- \* If asking a question, ask in public chat.
- \* If answering a question, answer in private chat.

Factor of a number
$$\frac{x}{y} = \text{integer} \Rightarrow y \text{ is a factor of } x.$$

$$24 \rightarrow 1 \quad 2 \quad 3 \quad 4 \quad 6 \quad 8 \quad 12 \quad 24$$

$$10 \rightarrow 1 \quad 2 \quad 5 \quad 10$$

$$4 \rightarrow 1 \quad 2 \quad 4$$

 $0 \rightarrow$  Given a number N, court the #factors of N. (N > 0)

smallest factor = 1

largest factor = N

Check if i is a factor of 
$$N \rightarrow (N\% i == 0)$$
 $f = 0$ 

for  $i \rightarrow 1$  to  $N\%$ 

if  $(N\% i == 0)$  # iterations =  $N$ 
 $f ++$ 

} return  $f$ 

If the system takes I sec for 10 8 iterations.

How much time it will take in above code for N=109?

# iterations = 
$$N = 10^9$$
 iterations  
=  $10 \times (10^8$  iterations)
$$= 10 \times (1 \text{ sec}) = 10 \text{ sec}$$

$$N = 10^{18} \rightarrow 10^{18} \text{ iterations}$$

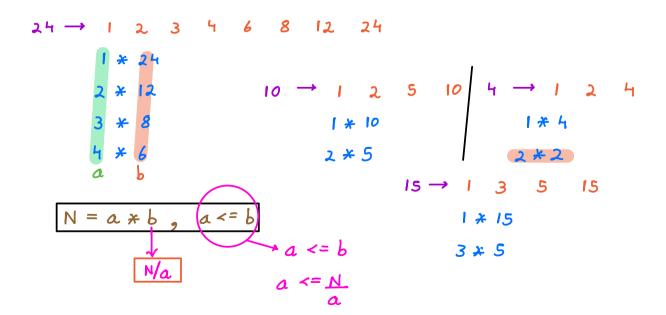
$$= 10^{10} \times (10^{8} \text{ iterations})$$

$$= 10^{10} \times (1 \text{ sec}) = 10^{10} \text{ sec}$$

$$\frac{10^{10}}{3600} \approx 2.7 \times 10^{6} \text{ hours}$$
  $\frac{10^{10}}{3600 \times 24} \approx 115741 \text{ days}$ 

$$\frac{10^{10}}{3600 * 24 * 365}$$
 =  $\frac{317 \text{ years}}{3600 * 24 * 365}$ 

Need to optimize!



$$a^2 \leftarrow N \Rightarrow \boxed{a \leftarrow \sqrt{N}}$$

amox = JN amin = 1 factors = 0 for  $a \rightarrow 1$  to  $\sqrt{N}$  or (a=1; a\*a <= N; a++)if (N% a == 0) { # iterations = N b = N/a if (a == b) factors += 1 / N → perfect square else factors += 2 I return factors

$$N = 10^{18} \rightarrow \sqrt{10^{18}}$$
 iterations =  $10^{9}$  iterations =  $10 * (10^{8}$  iterations) =  $10 * (18ec) = 108ec$ 

317 years --- 10 sec 6



10:32 PM

Prine Numbers & > 11, 23, 2, 31

+re numbers with exactly 2 factors.

if (court Factors (N) == 2) return true else return false

$$S = 1 + 2 + 3 + 4 + \dots + 99 + 00$$

$$+ \underline{S} = 100 + 99 + 98 + \dots + 101 \quad (100 \text{ times})$$

$$2S = 101 + 101 + 101 + \dots + 101 \quad (100 \text{ times})$$

$$2*S = 101 * 100$$

$$\Rightarrow S = \underline{101 * 100} = \underline{5050}$$

$$2$$

$$S = 1 + 2 + 3 + \dots + (N-1) + N$$

$$+ \underline{S} = N + (N-1) + \dots + 2 + 1$$

$$2*S = (N+1) + (N+1) + \dots + (N+1) \quad (N+1) \quad (N+1) \quad (N+1)$$

$$\Rightarrow 2*S = N*(N+1) \Rightarrow S = \underline{N*(N+1)}$$

[2 5] 
$$\rightarrow$$
 2 3 4 5  
[2 5)  $\rightarrow$  2 3 4 []  $\rightarrow$  includes boundary  
(2 5)  $\rightarrow$  3 4 ()  $\rightarrow$  excludes boundary  
[3  $10$ ]  $\rightarrow$  3 4 5 ...  $10 \rightarrow 8$   
[a b]  $\rightarrow$  b-(a-1) = b-a+1

# Fird # iterations

for 
$$i \rightarrow 1$$
 to  $N \notin \#iterations = N$ 

$$| if (i == N) \text{ break}$$
}

$$\begin{cases}
5 & \text{for } i \to 0 \text{ to } 100 \text{ for } 100 \text{ for } 100 = 100 \text{$$

3) for 
$$i \rightarrow 1$$
 to N of

if  $(i ? \cdot 2 = = 0)$ 

print  $(i)$ 

# iterations = N+M

for  $i \rightarrow 1$  to M of

if  $(i ? \cdot 2 = = 0)$ 

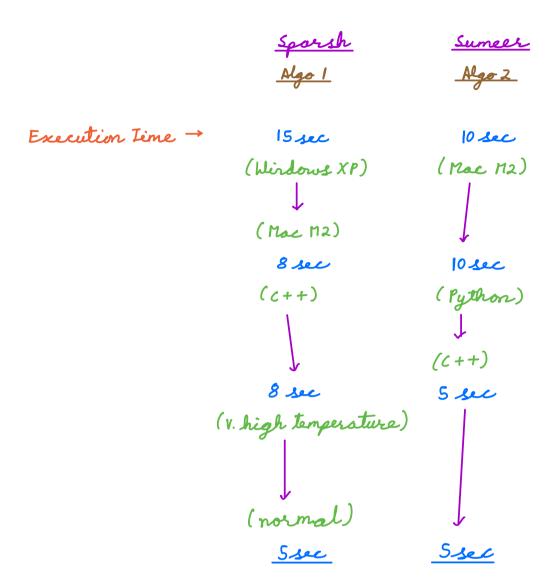
print  $(i)$ 

# Cometric Progression

$$S = a + a + x + a + x^{2} \dots (n \text{ terms})$$

$$S = \underbrace{a (x^{n} - 1)}_{(x - 1)} + \underbrace{H.W}_{H.W}$$

Story.
Arrange N numbers is ascerding order.



Moral → Execution time depards on multiple factors. ∴ better to use # iterations.

for 
$$i \rightarrow 1$$
 to  $N \notin \#iterations = N$ 

|  $i \neq (i == N) \text{ break}$ 
}