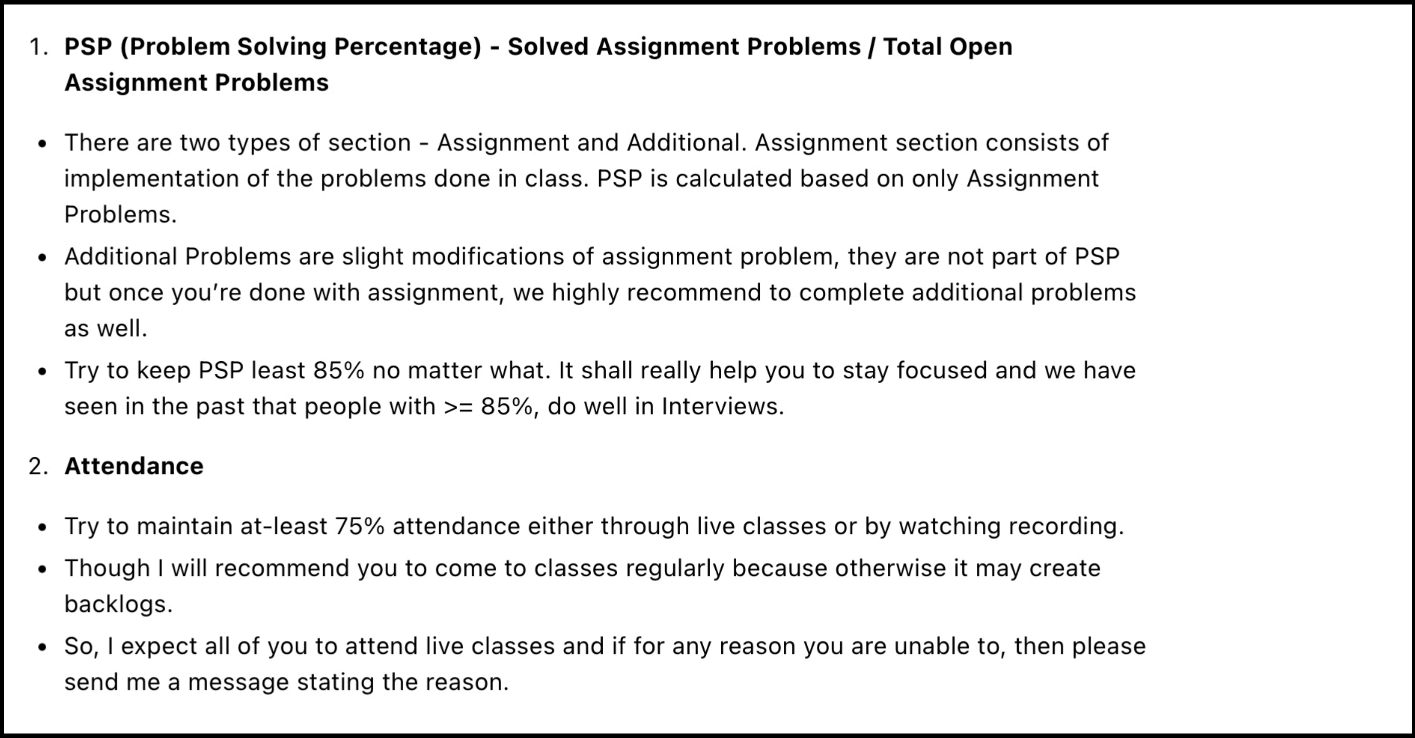
Suryansh Gupta

2021 Grad, SWE-III (L4) at Google

Interned at Mincosoft & Directi

Love Problem Solving was former Competitive Coder

IIEST Shibpur B.Tech in IT



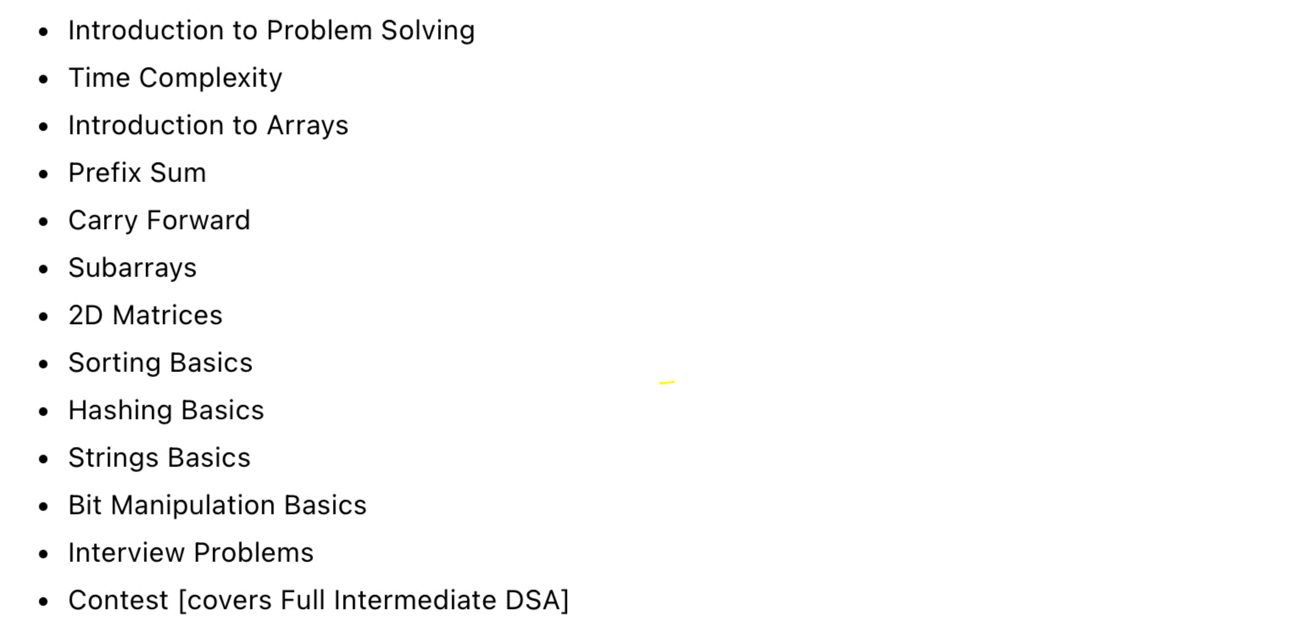
1. **PSP (Problem Solving Percentage) - Solved Assignment Problems / Total Open Assignment Problems**

- 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 85% no matter what. It shall really help you to stay focused and we have seen in the past that people with >= 85%, do well in Interviews.

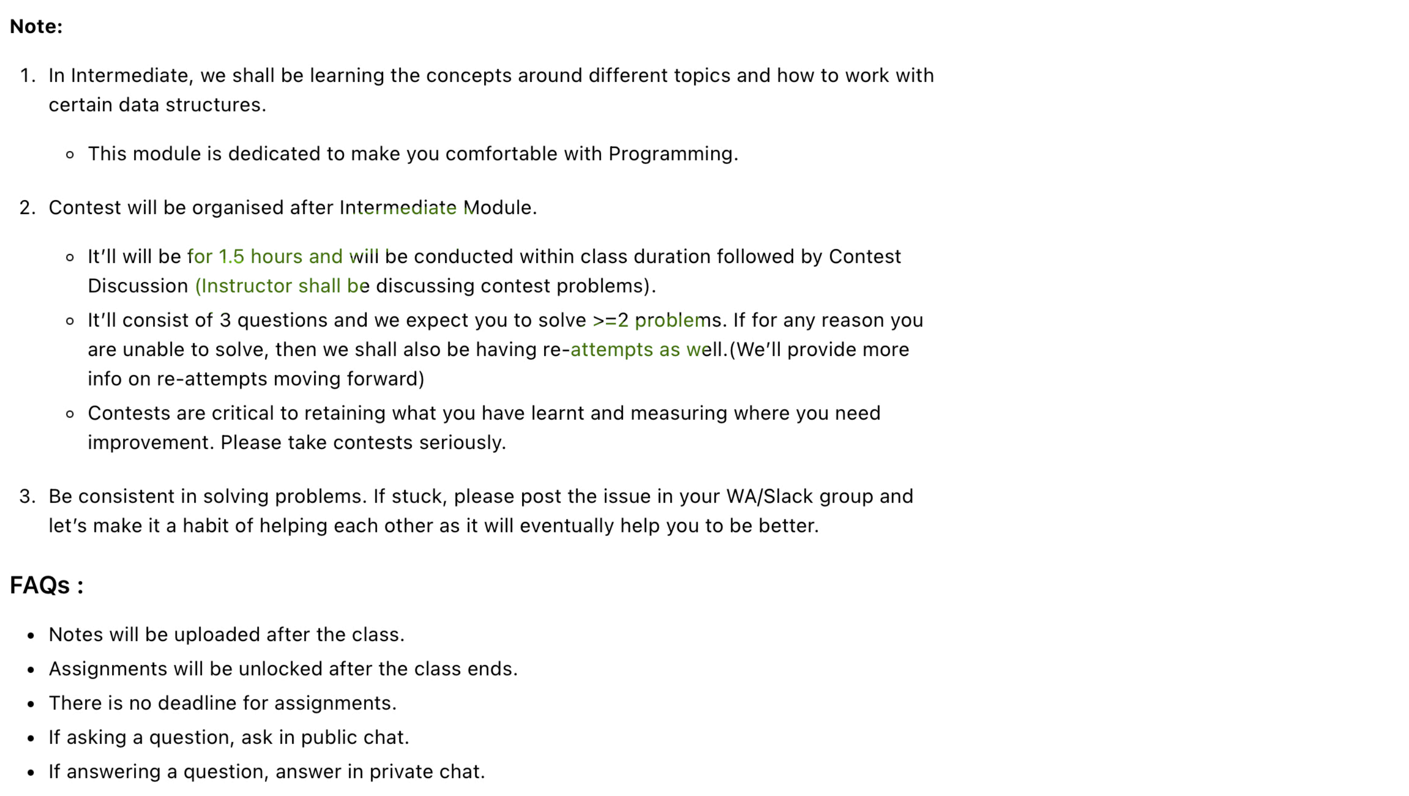
2. **Attendance**

- Try to maintain at-least 75% 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 me a message stating the reason.

Content for Intermediate Module



- 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]



Note:

1. In Intermediate, we shall be learning the concepts around different topics and how to work with certain data structures.
   - This module is dedicated to make you comfortable with Programming.
2. Contest will be organised after Intermediate Module.
   - It'll will be for 1.5 hours and will be conducted within class duration followed by Contest Discussion (Instructor shall be discussing contest problems).
   - It'll consist of 3 questions and we expect you to solve >=2 problems. If for any reason you are unable to solve, then we shall also be having re-attempts as well.(We'll provide more info on re-attempts moving forward)
   - Contests are critical to retaining what you have learnt and measuring where you need improvement. Please take contests seriously.
3. Be consistent in solving problems. If stuck, please post the issue in your WA/Slack group and let's make it a habit of helping each other as it will eventually help you to be better.

FAQs :

- Notes will be uploaded after the class.
- Assignments will be unlocked after the class ends.
- There is no deadline for assignments.
- If asking a question, ask in public chat.
- If answering a question, answer in private chat.

If you want to go fast, go alone,

If you want to go far, go together

Count of factors.

What is a factor ?

If any number devides N completely then they are factor of that number.

6 → 1, 2, 3, 6

24 → 1, 2, 3, 4, 6, 8, 12, 24 ⇒ 8 factors

10 → 1, 2, 5, 10 ⇒ 4 factors

Approach 1: Iterate in all numbers from 1 to N and check if i is a factor.

```
fac_count =
for i → 1 to N:
    if N % i == 0:
        fac_count = fac_count + 1
```

Code runs on severs

How much time it will take to ge output ?

1 GHz 10^8 iterations are done in 1sec.
‖
10^9 operations ⇒ 1 iteration ⇒ 10 op.

Iteration: No. of times a loop runs.

N	Iterations	Execution Time
10^8	10^8	1 sec.
10^9	10^9	10 sec. =>
10^{18}	10^{18}	

5 mango => Rs 10

50 mangos => Rs. 100

1 man => $\frac{10}{5}$

50 => $\frac{10}{5} \times 50$

= 100

$1 = \frac{1}{108}$ sec.

$10^{18} = \frac{10^{18}}{10^8}$ sec => 10^{10} sec.

=> 317 years.

You will be alive

⇓

Your children

⇓

2nd Generation

⇓

3rd Generation

⇓

4th Generation

24

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

Factors are repeting after some number.

$i \Rightarrow N/i$

Factors lies in pairs

what is the repetetion point.

$i > \frac{N}{i}$

$i^2 > N$

$i > \sqrt{N}$

after this there is repetetion

10

i	N/i
1	10
2	5
5	2
10	1

```
int count factors(N)
    fac_count =
    for i -> 1 to $\sqrt{N}$
        if N % i = = 0 :
            if( i ! = N / i)
                fac-count = fac-count + 2
            else
                fac-count = fac-count + 1
    return fac-count
```

100

i	N/i					
1	100	=>	+2			
2	50	=>	+2			
4	25	=>	+2			
5	20	=>	+2			
10	10	=>	+2	No	=>	+1

ans = 9

30

i	N/i	
1	30	+2
2	15	+2
3	10	+2
5	6	+2
6		

$N = 10^{18}$ Iterations $= \sqrt{10^{18}}$

$= 10^9$

Time = 10 sec.

10^8 iter = 1 sec.

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

10^9 iter = $\frac{10^9 \times 1}{10^8}$

= 10 sec.

317 years → 10 sec.

Q How many prime numbers are there?

10, 11, 23, 2, 25, 27, 31

Q What is a prime Number ?

A prime number is a number having exactly 2 factors.

1 ?

2 ? => YES 1, 2

```
boolean checkPrime(N) {
    if countfactors(N) == 2
        return True
    else
        return False
}
```

Game: Alok is a naughty $\xrightarrow{\qquad}$ Gauss

4th stand.

$$S = 1 + 2 + 3 \;.\;.\;.\;.\;.\;.\;.\;.\; 100$$

$$S = 100 + 99 + 98 \;.\;.\;.\;.\;.\;.\;.\; 2 + 1$$

$$2S = \underbrace{101 + 101 + 101 + 101 \;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\;.\; 101}_{100 \text{ times}}$$

$$2S = 100 * 101$$

$$S = \frac{100 * 101}{2} = 50 * 101$$

$$= 5050$$

$$S = 1 + 2 + 3 + \;.\;.\;.\;.\;.\;.\;.\; N-1 + N$$

$$S = N + N-1 + N-2 \;.\;.\;.\;.\;.\;.\;.\;.\; 2 + 1$$

$$2S = \underbrace{(N+1) + (N+1) + (N+1) + (N+1) \;.\;.\;.\;.\;.\;.\;.\; (N+1)}_{N \text{ times}}$$

$$2S = N * (N+1)$$

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

Ranges in Maths.

[→ inclusive of terminals

(→ excluding terminals.

[3 10] ⇒ 3 4 5 6 7 8 9 10

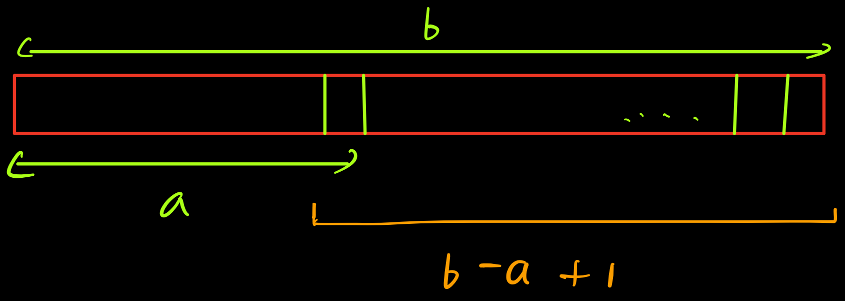
⇒ 8 numbers

(3 10] ⇒ 7 numbers

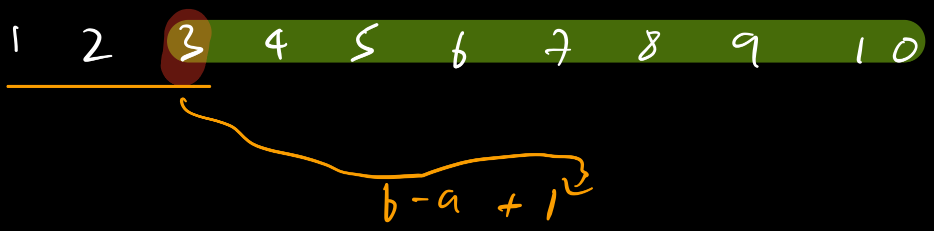
[3 10) ⇒ 7 numbers

(3 10) ⇒ 6 numbers

[a b] ⇒ $b - a + 1$



upperbound − lowerbound + 1



Iteration = Number of times loop runs.

Quiz 1:

```
for (i=0; i<=100; i++){

}
```

$[0, 100] \Rightarrow 100 - 0 + 1 = 101$

Quiz 2:

```
func() {
  for (int i=1, i<=N, i++){

  }
  for (int j=1, j<=M, j++){

  }
}
```

$[1 \quad N]$

$= N - 1 + 1 = N$

$[1 \quad M]$

$= M - 1 + 1 = M$

Ans: $N + M$

Geometric Progression

5 10 20 40 80 . . . .

↳ ×2 ↳ ×2 ↳ ×2

$\frac{10}{5} = 2$

$\frac{20}{10} = 2$

$\frac{40}{20} = 2$

$\frac{80}{40} = 2$

$a \quad ar \quad ar^2 \quad . . . . .$

a = First Term

r = common ratio

N^{th} Term = $Ar^{(n-1)}$

$$\frac{2(2^N-1)}{2-1}$$

Sum of first N terms of GP = $\frac{a(r^N-1)}{r-1}$

N = No. of terms. $= \frac{a(1-r^N)}{1-r}$

Quiz:

```
for (int i=1, i<=N, i++) {
    for (j=1, j<=(2^i), j++) {

    }
}
```

i	j: [1 2^i]	# iterations
1	[1 2]	2
2	[1 2^2]	4
3	[1 2^3]	8
4	[1 2^4]	16
⋮		
N	[1 2^N]	2^N

$a = 2$

$r = 2$

$N = N$ terms.

$$\frac{2(1-2^N)}{1-2} = \frac{2(1-2^N)}{-1}$$

$$\Rightarrow 2(2^N-1)$$

1 5 25 125 625

$r = 5$

$$1 * 5^{N-1}$$

128 64 32 16 8 4

$r = 1/2$

$$A_n = 124 * \left(\frac{1}{2}\right)^{N-1}$$

$$\frac{64}{128} = \frac{32}{64} = \frac{16}{32} = \frac{8}{16} = \frac{4}{8} = 1/2$$

Q Game: I gave a problem to develop a sorting algorithm to sort given numbers.

[5 10 1 3 2] => [1 2 3 5 10]

Hiresh

15 sec.

Windows XP - Pentium

↓

Macbook m2

7 sec.

C++

I am at a Top of a Volcano

↓

Mount Everest

5 sec.

Ganesh

10 sec.

Mac m2

↓

Python

↓

C++

5 sec.

Mount Everest

We can't evaluate algorithm's performance using execution time it depends on lot of factors.

→ OS, → Machine, → Place of execution, language.

Next Class:

- We will start with Big O notations.
- Explore Logarithm
- See Space Complexity
- End by TLE & importance of constraints

Doubts

AP

2 4 6 8 10 12 . . . .

a a+d a+2d a+3d . . . .

$a = 2$ a = first Term

$d = 2$ d = common difference.

Execution Speed

C++	Java	Python
1se.	1.5 - 2 sec.	3 sec.

Given N, give the sum of first N natural numbers.

$$\frac{N * (N+1)}{2}$$