Introduction to Problem Solving

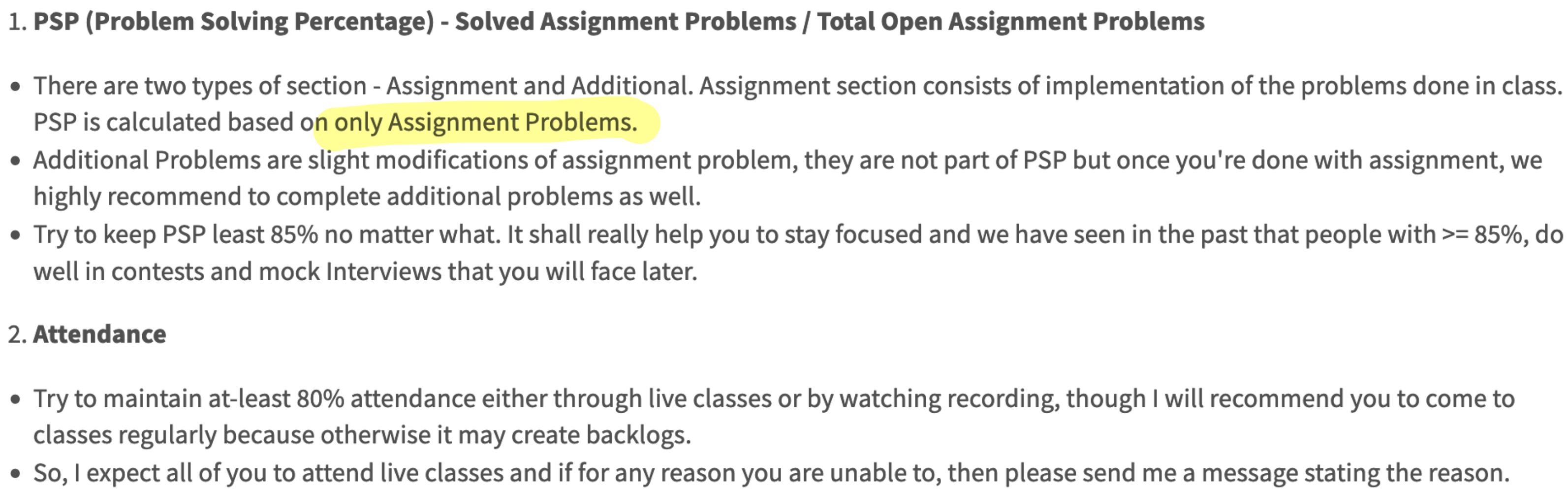
Hello Everyone :)

→ Welcome to intermediate module of DSA

→ Nipun Mittal Googler

→ BTech - 2019, MTech - 2021 (IIT Bombay)

→ ~3 years of part-time teaching experience

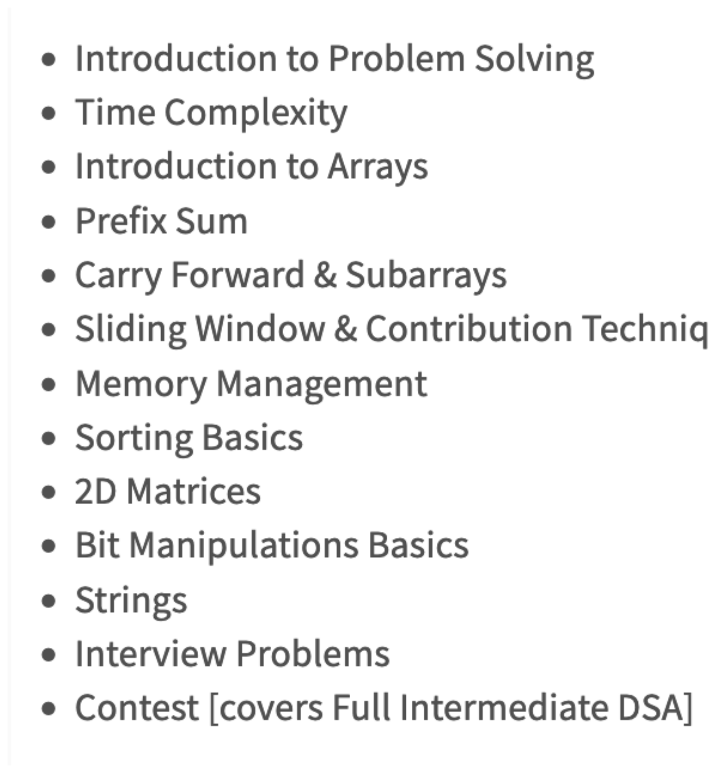


1. **PSP (Problem Solving Percentage) - Solved Assignment Problems / Total Open Assignment Problems**

- 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 contests and mock Interviews that you will face later.

2. **Attendance**

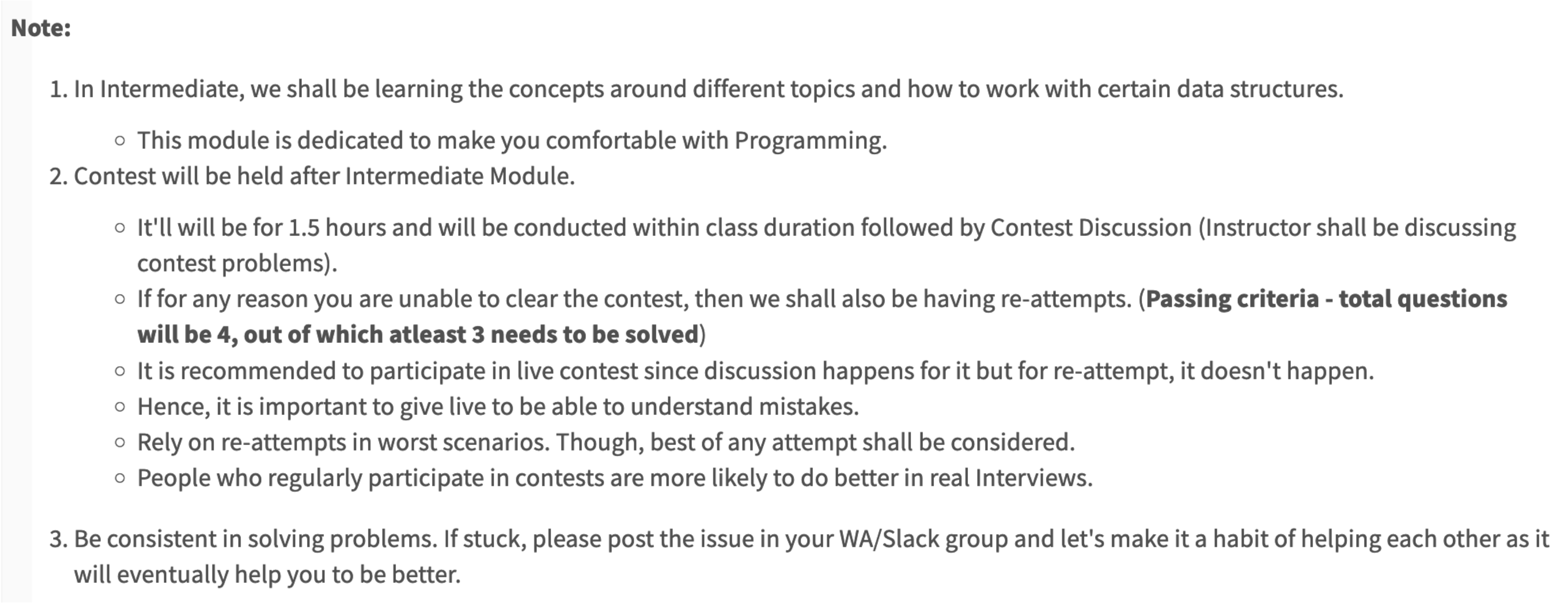
- Try to maintain at-least 80% attendance either through live classes or by watching recording, though I will recommend you to come to classes regularly because otherwise it may create backlogs.
- So, I expect all of you to attend live classes and if for any reason you are unable to, then please send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



- Introduction to Problem Solving
- Time Complexity
- Introduction to Arrays
- Prefix Sum
- Carry Forward & Subarrays
- Sliding Window & Contribution Technique
- Memory Management
- Sorting Basics
- 2D Matrices
- Bit Manipulations Basics
- Strings
- Interview Problems
- Contest [covers Full Intermediate DSA] → 1.5 hrs

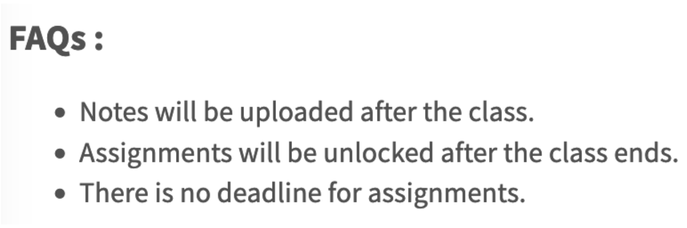
4 questions

↓ 1/3 of them to clear



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 held 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).
    - If for any reason you are unable to clear the contest, then we shall also be having re-attempts. (**Passing criteria - total questions will be 4, out of which atleast 3 needs to be solved**)
    - It is recommended to participate in live contest since discussion happens for it but for re-attempt, it doesn't happen.
    - Hence, it is important to give live to be able to understand mistakes.
    - Rely on re-attempts in worst scenarios. Though, best of any attempt shall be considered.
    - People who regularly participate in contests are more likely to do better in real Interviews.
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.

1. Count the factors
2. Optimization for count the factors
3. Check if a number is prime
4. Sum of N natural no.
5. Number of iterations ?
6. How to compare two algos ?

What is a factor ?

i is a factor of N if i divides N completely

$N \% i == 0$

Given N, count the factors of N.

$N > 0$

24 → 1, 2, 3, 4, 6, 8, 12, 24   ans = 8

10 → 1, 2, 5, 10   ans = 4

smallest factor = 1

largest factor = N

=) all factor lies between 1 to N.

```
count factors (N) {
    ans=0
    for(i=1  to N) {
        if(N%i ==0)
            // i is a factor of N
            ans++
    }
```

Assume :

10^8 iterations per sec

1 iteration → $\frac{1}{10^8}$ sec

N iterations → $\frac{N}{10^8}$ sec

```
|   return ans
}
```

above code takes N iterations

say, $N = 10^9 \Rightarrow \frac{N}{10^8}$ sec $= \frac{10^9}{10^8} = 10$ sec

$N = 10^{18} \Rightarrow \frac{10^{18}}{10^8}$ sec $= 10^{10}$ sec ~ 317 years

you → children → 3rd gen → . . .

Optimization

if $i \times j = N \Rightarrow i$ & j are factors of N

$j = N/i \Rightarrow \{i, N/i\}$ are factors of N

$$\boxed{\begin{array}{c}\text{if } i \text{ is a factor of } N \\ \text{then } N/i \text{ is also a factor of } N\end{array}}$$

N = 24

i		N/i		count of factors ↑ c = 0
1	<	24	24/1	c=c+2
2	<	12	24/2	c=c+2
3	<	8	24/3	c=c+2
4	<	6	24/4	c=c+2
				c = 8
6	>	4	24/6	
8	>	3	24/8	
12	>	2	24/12	
24	>	1	24/24	

N = 100

i		N/i		c=0
1	<=	100	100/1	c=c+2
2	<=	50	100/2	c=c+2
4	<=	25	100/4	c+=2
5	<=	20	100/5	c+=2
10	<=	10	100/10	c++ ~~c+=2~~
20		5	100/20	~~c=10~~ c=9
25		4		
50		2		
100		1		

Iterate till

$i <= N/i$

$i*i <= N \Rightarrow i^2 <= N \Rightarrow i <= \sqrt{N}$

$i = [1 \ldots\ldots \sqrt{N}]$ $\sqrt{N}$ iterations

```
count factors(N) {
    c=0              i*i<=N  → use this in code
                       ↑
    for(i= 1; i<=√N;  i++){
        if(N%i ==0) {
            // i is a factor of N & N/i is also factor of N
            if (i == N/i)
                c++
            else
                c += 2
        }
    }
    return c
```

for N = 10^{18} ⇒ $\sqrt{N}$ iterations

$\sqrt{10^{18}} = 10^{9}$ iterations

10^{8} iterations → 1 sec

10^{9} iterations → 10 sec

N=24 $(i*i <= 24)$

c=0

i=1   c=c+2   c=2

i=2 $(2*2<=24)$ c=c+2 = c=4

i=3 $(3*3<=24)$ c=c+2 = c=6

i=4 $(4*4<=24)$ c=c+2 = c=8

i=5 $(5*5 \cancel{\le} 24)$

Given N, Check if N is prime or not.

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

ans=4

Prime Number

1. Number divisible by 1 & itself. ✗

2. Has exactly 2 factors ✓

N=1 [not prime]

```
isPrime(N) {
    if (countfactors(N) == 2) {
        return true
    }
    return false
}
```

$\sqrt{N}$ iterations

Some basic math properties

$[a, b]$ → this type of range means that a and b both are inclusive.

(a, b) → this type of range means that a and b both are excluded.

$[3, 10]$ → 3, 4, 5, 6, 7, 8, 9, 10

count=8

$(2, 5)$ → 3, 4

count=2

$[2, 5]$ → 3, 4, 5

count=3

no.s b/w $[a, b]$?

$[a, b]$ → $a, a+1, a+2, \ldots\ldots, b$

$\Rightarrow$ $b - a + 1$

(a, b) → $b - a - 1$

$[a, b)$ → $b - a$

Story of a 4th class boy (Gauss)

find the value of

$$1 + 2 + 3 + \dots\dots + 100$$

$$S = 1 + 2 + 3 + \dots\dots + 99 + 100$$

$$S = 100 + 99 + 98 + \dots\dots + 2 + 1$$

$+$

$$2S = 101 + 101 + 101 + \dots\dots + 101 + 101$$

$$2S = 101 \times 100$$

$$S = \frac{101 \times 10\cancel{0}}{\cancel{2}} = 101 \times 50 = 5050$$

Find the sum of first N natural numbers?

$$S = 1 + 2 + 3 + \dots\dots + N$$

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

What is iterations ?

Number of times a loop runs

```
for( i = 1 to N) {
    if (i==N)  break;
}
```

$[1, N]$

$\Rightarrow b-a+1$

$\Rightarrow N-1+1 = N$

```
for( i=0 to 100) {
    s = s + i + i^2;
}
```

$[0, 100]$

$\Rightarrow 100-0+1 = 101$

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

$\rightarrow [1, N] \Rightarrow N$ iterations

$\rightarrow [1, M] \Rightarrow M$ iterations

total iterations $= N + M$

Geometric Progression (GP)

$$\boxed{5}, \quad 10, \quad 20, \quad 40, \quad 80, \quad \ldots\ldots$$

first term (a)

$10/5 = \boxed{2}$ $\quad \frac{20}{10} = \boxed{2}$ $\quad \frac{40}{20} = \boxed{2}$ $\quad \frac{80}{40} = \boxed{2}$

common ratio (r)

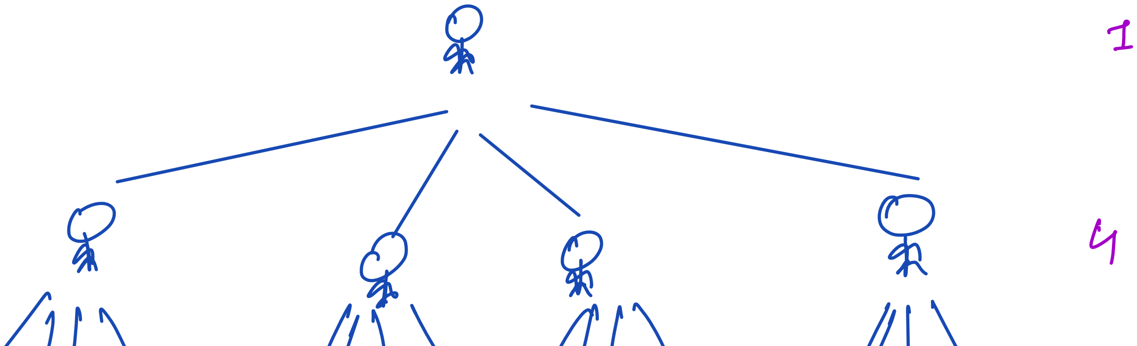
$$a, \quad ar, \quad ar^2, \quad ar^3, \ldots\ldots\ldots$$

n terms : $a, \; ar, \; ar^2 \; \ldots\ldots\ldots \; ar^{n-1}$

1st 2nd 3rd nth

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

Suppose one person spread a news to 4 diferent persons. Each of those, spread to 4 others & so on . . . .



4^2

⋮

4^3

⋮

8th level

$1 \times 4^{8-1}$

$= 4^7 \Rightarrow 16384$

Sum till 8th level $= \frac{a(r^n - 1)}{r-1}$

$$\frac{1(4^8 - 1)}{4-1} = \frac{65536 - 1}{3} = 21845$$

Jatin

Algo 1

15 sec

↓

(Windows XP)

↓

(macbook M3)

↓

Faraan

Algo 2

10 sec

↓

(macbook M3)

↓

compare their execution time ?

7 sec

↓

(C++)

↓

7 sec

(Top of hot Volcano)

↓

(Mt. Eversest)

5 sec

⋮

10 sec

↓

(Python)

↓

(C++)

5 sec

(Mt. Everest)

↓

5 sec

⋮

We can't judge 2 algos based on execution time because it depends on lot of factors like OS, place of execution, language, etc.

How can we compare 2 algos?

→ Number of iterations

Next Clam

→ Big O

→ logarithms

→ space complexity

→ TLE error & importance of constraints.