Agenda :

1. Introduction
2. Module Description
3. Count Of Factors
4. Check if a number is prime
5. Sum of 1st N natural numbers.
6. No. of iterations
7. Comparing Algorithms.

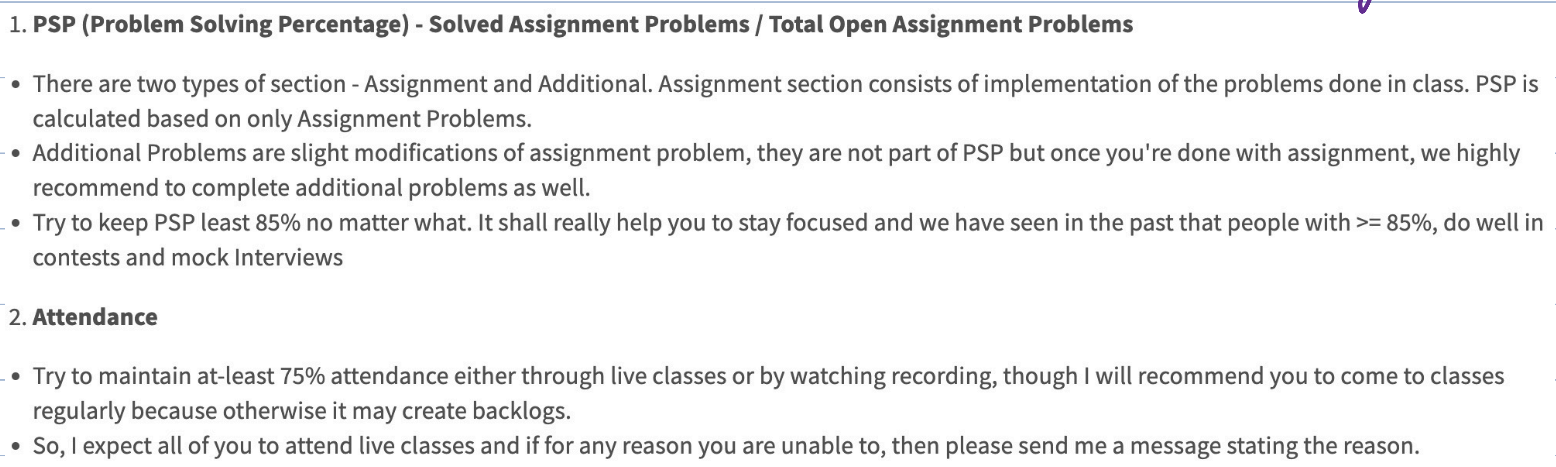
Introduction:

Name : Harsha

From : Udupi

→ Maersk

→ SJCE Mysore



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

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.

Approach:

1 → Watch recorded sessions at higher speeds
2 → Assignments
3 → Additional Problems.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
7-9.30 AM	✓	✓ P	✓	✓ P	✓	✓ P	R

Module Description

Time Complexity
Arrays
Prefix Sum
Carry Forward
Subarrays.
Matrices
Sorting Basics.
Hashing Basics.
Strings Basics.
Bit Manipulation Basics.
Interview Problems.
Contest

Objective : 1. Comfortable with writing programs.

Contests :

Points To Remember :

1
2.
3.

Question :

Given N, return the count of factors of N.
$N \geq 0$

$[1, N]$

What is a factor?

If x is a factor of $N \Rightarrow N \% x == 0$

Factors of 36 → 1 2 3 4 6 9 12 18 36 → 9

Factors of 24 → 1 2 3 4 6 8 12 24

Factors of 100 → 1 2 4 5 10 20 25 50 100 → 9

Approach 1:

```
int countFactors(int N)
{   int count = 0;
    for (i → 1 to N)
    {  if(N % i == 0)
         { count ++;
         }
    }
    return count;
}
```

HW → Dry run this for 36 , 16

N=8

i	count(0)
~~1~~	~~1~~
~~2~~	~~2~~
~~3~~	~~2~~
~~4~~	~~3~~
~~5~~	~~3~~
~~6~~	~~3~~
~~7~~	~~3~~
~~8~~	4

Observations :

No. of iterations that a single core CPU can execute in one second ≃ 10^8

N	$\sqrt{N}$	Iterations	Execution Time
8	2	8	0.0000 ...
100	10	100	0.000001 s
10^9	10^4	10^9	10 secs. 0.0001
10^{18}	10^9	10^{18}	10^{10} sec. 10 seconds

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

$= 8/10^8 \simeq$

$1000000 = 10^6$

$1000000000 = 10^9$

$\frac{100}{10^8} = \frac{1}{10^6}$

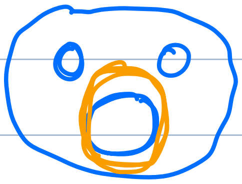
$\frac{10^9}{10^8} = 10$

$\frac{10^{18}}{10^8} = 10^{10}$ secon

$10^{9/2} \simeq 10^4$

$\frac{10^4}{10^8} = 10^{-4} = 0.0001$

317 years. → 10 sec



Optimisation :

$i=1 \; ; i \times i \leq N \quad ; i{+}{+}$

$i \leq \sqrt{N}$

$i \times i \leq N$

```
int   countFactors ( int  N)
{   int  count =0;
    for ( i=1 ;  i x i ≤ N : i++)
    {  if(N% i==0)
       {  if( i== N/i) {count++; }
          else { count = count +2; }

       }
    }
    return count;
}
```

	36	
1	36	→ 2
2	18	→ 4
3	12	→ 6
4	9	→ 8
6	6	→ 10
9	4	
12	3	
18	2	
36	1	

	63
1	63
3	21
7	9
9	7
21	3
63	1

$\sqrt{N} \to 7$

$\sqrt{24} = 4$

	24
1	24
2	12
3	8
4	6
6	4
8	3
12	2
24	1

Break : till 8:53 AM

Question:

Given a no. N, check if N is prime or not.

Definition: Any number with exactly 2 factors.

31 → 1 and 31

7 → 1 and 7

3 → 1 and 3

HW

$N \rightarrow \underline{\underline{1 \text{ to } N}}$

$N = 100.$

$S = 1 + 2 + 3 + 4 + 5 + \ldots\ldots + 100$

$S = 100 + 99 + 98 + 97 + 96 + \ldots\ldots + 1$

N

$2S = (100+1) + (99+2) + (98+3) + (97+4) + \ldots\ldots + (1+100)$

$= (N+1) * N$

$2S = N(N+1)$

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

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

$$\frac{100(101)}{2} = 50 \times 101 = \underline{\underline{5050}}$$

$[1, N] \rightarrow (N-1+1) \rightarrow N$ iterations.

$[0, 100] \rightarrow (100-0+1) \rightarrow \underline{\underline{101}}$

Basic Math:

1. $[a,b] \rightarrow b-a+1$
2. $(a,b) \rightarrow b-a-1$

$\boxed{[a,b]} \rightarrow$ Both a & b

$[a,b) \rightarrow$ a but not b

$(a,b) \rightarrow$ excluding a & b

$[3,10] \rightarrow 3,4,5,6,7,8,9,10$

↳ $10-3+1$

Iterations:

$[1,N] \rightarrow (N-1+1) \rightarrow$ N iterations.

$[0,100] \rightarrow (100-0+1) \rightarrow \underline{\underline{101}}$

$i \rightarrow (1 \rightarrow N) \rightarrow N$

$j \rightarrow (1 \rightarrow M) \rightarrow M$

$\underline{\underline{N+M}}$.

Progressions :

$$1 \to 2 \to 4 \to 8 \to 16 \to 32$$

$4/2 = 2$ $\qquad$ $CR = r = 2$

$16/8 = 2$

$\underset{a}{3} \quad 9 \quad 27 \quad 81 \quad 243$

$r = 3$

n terms

$$\underbrace{\overset{a}{3} + 9}_{r} + 27 + 81 + 243 \dots$$

$$\begin{array}{r} 3 \\ 9 \\ 27 \\ 81 \\ \hline 120 \end{array}$$

$$S_n = \frac{a(r^n - 1)}{(r - 1)}$$

$a = 5$

$r = 3$

$$S_4 = \frac{3(3^4 - 1)}{2} = \frac{3 \times (80)}{2}$$

$$= 3 \times 40 = 120.$$

Comparing Algorithms:

Virendra

32gb ram M2 chip 5

50°C

18°C 3·5s

Prashant

10 XP

3·5s 32gb ram M2 chip

18°C

Moral : Use # of iterations → Time Complexity.

NOT Execution Time

Next Class Content :

- Big O
- Logarithm
- TLE & Importance of Constraints