



Hello World !

Introduction

FAQ's

## Introduction

### Education

B.Tech In Computer Science  
Scaler

### Work experience

SDIII at bsense

SSE at Dell (2022)

### Hobbies

Competitive Programming

Travelling

Origami

Few terms you will see or hear in 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 85% no matter what. It shall really help you to stay focused and we have seen in the past that people with  $\geq 85\%$ , 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.

## Intermediate Module Discussion

1. Introduction to Problem Solving
2. Time Complexity
3. Introduction to Arrays
4. Prefix Sum
5. Carry Forward
6. Subarrays
7. 2D Matrices
8. Sorting Basics
9. Hashing Basics
10. Strings Basics
11. Bit Manipulation Basics
12. Interview Problems
13. Contest [ Full Intermediate DSA ]

Make you  
comfortable with  
programming !

Tests what you retain

## FAQ's

Notes will be uploaded after the class

Assignments will be unlocked after class ends

Got a question ? Ask in Public

Got an answer ? Answer in Private

## Agenda

Count the factors

Check if a number is Prime

Sum of N natural numbers

Number of iterations

Comparing two algorithms

What is a factor?

i is a factor of N  $\frac{N}{i} = 0$  as remainder

% modulo operator

$N > 0$

Number of factors for  $N = 24$

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

Number of factors for 10.

10 - 1, 2, 5, 10

Given a number  $N > 0$ , return count of factors

Brute force

least factor - 1

highest factor -  $n$

Brute force

factor = 0

for  $i = 1 \rightarrow N$

if  $(N \% i == 0)$

factor ++

print (factor)

$n$  times

Any given server will run  $10^8$  iteration  
in 1 second

N	Iteration	Sec
$10^8$	$10^8$	1
$10^9$	$10^9$	$\frac{10^9}{10^8} = 10$ sec

$10^{18}$

$10^{18}$

$$\frac{10^{18}}{10^8} = 10^{10} \text{ sec}$$

$$10^{10} = \frac{317 \text{ years}}{4 \text{ generation}}$$

## Optimization

$$i \times j = N \quad (\text{factor occur in pair})$$

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

$$N = 24$$

repeating factors

① Top half has all factors

② Factors are repeating beyond a point

$$N = 100$$

$$i \times j = 100$$

$i$	$N/i$
1	100
2	50
4	25
5	20
10	10
20	5
...	...

repeating beyond this point

$$i \leq \frac{N}{i} = i \times i \leq N$$

$$i \leq \sqrt{N}$$

## pseudo code

```
def count_factors (n) {
```

```
    factor = 0
```

```
    for (i = 1; i * i <= N; i++) {
```

```
        if N % i == 0 {
```

```
            if i == N / i    factor += 1
```

```
            else    factor += 2
```

```
    return factor
```

}

$\sqrt{n}$

Iteration

N	iterations	secs
$10^{18}$	$\sqrt{10^{18}} = 10^9$	$\frac{10^9}{10^8} = 10 \text{ seconds}$

317 years  $\longrightarrow$  10 seconds

power of observation

## Quiz

List down all prime Numbers

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



Number having exactly 2 factors

Question

Given  $n > 0$ , check if it is prime or not.

```
if count_factors(n) == 2 {  
    |    print("Yes");  
}  
  
else {  
    |    print("No");  
}
```

---

Gauss

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

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

---

$$2S = \underline{101 + 101 + 101 + \dots + 101}$$

↳ occurring 100

$$2S = 101 * 100$$

$$S = \frac{101 * \cancel{100}^{50}}{\cancel{2}} = 5050$$

Generalize Sum of first N natural numbers

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

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

---


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


---

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

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

↑  
sum of first N natural  
numbers

10:18 → 10:25 pm

## Basic Maths

$[a, b]$  - range is inclusive of both  
a & b

$(a, b)$  - range excluding a and b

example

# values in range  $[3, 10]$

3, 4, 5, 6, 7, 8, 9, 10 — 8 number

Generalize

# values in  $[a, b]$

a - lower bound

b - upper bound

both are

included

# values in range  $b - a + 1$

$$[3, 10] = 10 - 3 + 1 = 8$$

## Iterations

number of times a loop runs

## # iteration

```
for (i = 1 ; i <= N ; i++) {  
    |     if (i == N) break  
    |  
}
```

Both

$[1, N]$

are included

$[1, N]$  range

$$\begin{aligned}\text{iteration} &= N - 1 + 1 \\ &= N\end{aligned}$$

## example

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

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

= 101 iterations

## example

```
for ( i = 1 ; i <= N ; i++ ) {  
    |      // do something  
}
```

N Iterations

```
for ( i = 1 ; i <= M ; i++ ) {  
    |      // do something  
}
```

M Iterations

$$\# \text{ Iteration} = N + M$$

---

## Geometric Progression

5 , 10 , 20 , 40 , 80 , 160 , 320

Sequence which has a common ratio  
multiplication

Generic Notation

$a, ar, ar^2, ar^3, \dots$

Sum of first N terms in GP

first term = a

Common ratio = r

n terms

$r \neq 1$

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

Home work

problem

Sort a given sequence of array  
with length of array as  $10^8$

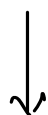
Tammy Bhat

(Algo 1)



10 sec

Macbook M3



10 seconds

(python)



Kullu

(Algo 2)



15 seconds

Windows XP



7 seconds

Macbook M3

(c++)



↓  
5 seconds

C++

↓  
7 seconds

1000 parameters that determine execution time

execution time not the correct metric to verify efficiency of code

No. of iteration remains same irrespective of parameter