***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

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

24 → 1 2 3 4 6 8 12 24

10 → 1 2 5 10

4 → 1 2 4

Q → Given a number N, count the #factors of N.
(N > 0)

smallest factor = 1

largest factor = N

check if i is a factor of N → (N % i == 0)

```
f = 0
for i → 1 to N {
    if (N % i == 0)
        f++
} return f
```

#iterations = <u>N</u>

If the system takes 1 sec for $10^8$ iterations.
How much time it will take in above code for $N = 10^9$?

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

$$\boxed{x^{a+b} = x^a * x^b}$$

$$N = 10^{18} \rightarrow 10^{18} \text{ iterations}$$
$$= 10^{10} * (10^8 \text{ iterations})$$
$$= 10^{10} * (1 \text{ sec}) = \underline{10^{10} \text{ sec}}$$

$$\frac{10^{10}}{3600} \approx 2.7*10^6 \text{ hours} \quad \Big| \quad \frac{10^{10}}{3600*24} \approx 115741 \text{ days}$$

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

Need to optimize!

24 → 1 2 3 4 6 8 12 24

| a | | b |
|---|---|---|
| 1 | * | 24 |
| 2 | * | 12 |
| 3 | * | 8 |
| 4 | * | 6 |

10 → 1 2 5 10

1 * 10

2 * 5

4 → 1 2 4

1 * 4

2 * 2

15 → 1 3 5 15

1 * 15

3 * 5

$$\boxed{N = a * b, \quad a <= b}$$

$$\boxed{N/a}$$

$$a <= b$$

$$a <= \frac{N}{a}$$

$50 = 5 * 10$    $a^2 <= N \Rightarrow \boxed{a <= \sqrt{N}}$

$a_{min} = 1$    $a_{max} = \sqrt{N}$

```
factors = 0
for a → 1 to √N {     // for (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
    }
} return factors
```

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

$= 10 * (10^8$ iterations$)$

$= 10 * (1$ sec$) = \underline{10 \text{ sec}}$

$\boxed{317 \text{ years} \longrightarrow 10 \text{ sec}}$ 😮

$\boxed{10:32 \text{ PM}}$

---

<u>Prime Numbers</u>    Eg → 11, 23, 2, 31

+ve numbers with exactly 2 factors.

```
if (countFactors (N) == 2)  return true
else      return false
```

---

$$S = 1 + 2 + 3 + 4 + \dots + 99 + 100$$

$$+\ \underline{S = 100 + 99 + 98 + \dots \quad 2 + 1}$$

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

$$2 * S = 101 * 100$$

$$\Rightarrow S = \frac{101 * 100}{2} = \underline{5050}$$

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

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

$$2 * S = (N+1) + (N+1) + \dots \quad (N+1) \quad \text{(N times)}$$

$$\Rightarrow 2 * S = N * (N+1) \rightarrow \boxed{S = \frac{N * (N+1)}{2}}$$ ✓

---

[2 5] → 2 3 4 5

[2 5) → 2 3 4

(2 5) → 3 4

[ ] → includes boundary

( ) → excludes boundary

[3 10] → 3 4 5 . . . 10 → $\underline{8}$

[a b] → $b - (a-1) = \underline{b - a + 1}$

---

## Find # iterations

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

# iterations = $\underline{N}$

```
2)  for i → 0 to 100 {
        s = s + i + $i^2$
    }
```

[0  100] → 100 - 0 + 1 = 101

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

# iterations = N + M

## Geometric Progression

2   6   18   54 . . . .

$r = 3$

$S = a + a*r + a*r^2 . . .$ (n terms)

$$\boxed{S = \frac{a(r^n - 1)}{(r - 1)}} \quad \boxed{r \, != 1}$$

→ H.W

## Story

Arrange N numbers is ascending order.

| | Sparsh | Sumeer |
|---|---|---|
| | Algo 1 | Algo 2 |
| Execution Time → | 15 sec | 10 sec |
| | (Windows XP) | (Mac M2) |
| | ↓ | ↓ |
| | (Mac M2) | |
| | 8 sec | 10 sec |
| | (c++) | (Python) |
| | ↓ | ↓ |
| | | (C++) |
| | 8 sec | 5 sec |
| | (v. high temperature) | |
| | ↓ | ↓ |
| | (normal) | |
| | 5 sec | 5 sec |

Moral → Execution time depends on multiple factors.

∴ better to use # iterations.

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

#iterations = N ✓