

Welcome 😊

FAQs

- 1) Notes will be uploaded
- 2) PSP → Problem Solving Percentage. ($> 90\%$) ⇒ assignment problems.
- 3) Assignments & Additional Problems
- 4) Attendance ⇒ ($> 75\%$)
- 5) Contest (1-2)
1:30 hr.

6) Intermediate Topics

- 1) Intro.
- 2) Time Complexity.
- 3) Arrays (5) ✓
- 4) Sorting. ✓
- 5) Hashing. ✓
- 6) Strings. ✓
- 7) Bit manipulation. ✓
- 8) Interview problems.
- 9) Contest.

⇒ Question → public chat
Answer → private chat

Agenda:

- Count of factors.
- Optimise
- Prime No.
- Basic math.
- Iterations.
- Compare.

Q1 Given a number N , count the factors of N .

factors \Rightarrow any number i which divides N completely
is a factor of N

$$N \% i == 0$$

$$24 = \{1, 2, 3, 4, 6, 8, 12, 24\} \quad \underline{\underline{\text{Ans} = 8}}$$

$$10 = \{1, 2, 5, 10\} \quad \underline{\underline{\text{Ans} = 4}}$$

Pseudo code

```
int countFactors (int N)
{
    int factors = 0
    for( i  $\rightarrow$  1 to N)
    {
        if (  $N \% i == 0$  )
            factors ++
    }
    return factors
}
```

1) execution time
 \rightarrow input
 \rightarrow iterations
 \rightarrow system

assumption : 10^8 iterations \approx 1 sec

N	iterations	execution time.
10^8	10^8	<u><u>1 sec</u></u>
10^9	10^9	<u><u>10 sec</u></u>
10^{18}	10^{18}	<u><u>10^{10} sec</u></u> \approx 317 years

Optimize

$$i * j = N \Rightarrow i \text{ \& } j \text{ are factors of } N$$

$$j = \frac{N}{i} \Rightarrow i \text{ \& } \frac{N}{i} \text{ are factors of } N$$

eg: 24

i		N/i	
1	\leq	24	P ₁
2	\leq	12	
3	\leq	8	
4	\leq	6	
6	\geq	4	P ₂
8	\geq	3	
12	\geq	2	
24	\geq	1	

eg: 100

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

obs 1 after certain no., factors are repeating

obs 2

$$i^0 \leq \frac{N}{i^0}$$

$$i^2 \leq N$$

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

```
int countFactors (int N)
```

eg: 100

```
{ int factors = 0
```

```
for( i = 1 ; i * i ≤ N ; i++)
```

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

√N iterations

```
{ if ( i == N/i ) } perfect square.
```

```
    factors += 1
```

```
    else
```

```
        factors += 2
```

```
}
```

```
return factors
```

```
}
```

N

iteration

operation

10^{18}

10^9

10 sets

10:35

Q Given N, you need to check if it is prime or not?

⇒ { 10, 11, 23, 2, 25, 27, 31 } ans=4

X

✓

✓

✓

X

X

✓

⇒ Prime numbers only have 2 factors ⇒ 1 & N

Pseudocode

```
bool checkPrime (int N)
```

```
{ if ( countFactors(N) == 2)
```

```
    return True
```

```
    else
```

```
        return False
```

```
}
```

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

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

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

$$2S = 100 \times 101$$

$$S = \frac{100 \times 101}{2}$$

Sum of first N natural numbers

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

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

$[a, b]$ \leftarrow a & b both are inclusive.

(a, b) \leftarrow a & b both are excluded.

eg: $[3, 10] \Rightarrow \underline{3}, 4, 5, 6, 7, 8, 9, \underline{10} \Rightarrow 8.$

$$[a, b] \Rightarrow b - a + 1$$

Iteration

\Rightarrow Number of times loop runs c/a Iterations.

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

```
{
    if ( i == N )
        break
}
```

Aus = N

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

Ans =

101

[0, 100]

$$100 - 0 + 1 = 101$$

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

[1, N]
⇒ N

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

Ans = N + M

[1, M]
⇒ M

Geometric Progression

5 10 20 40 80

a → first term → 5

r → common ratio → 2

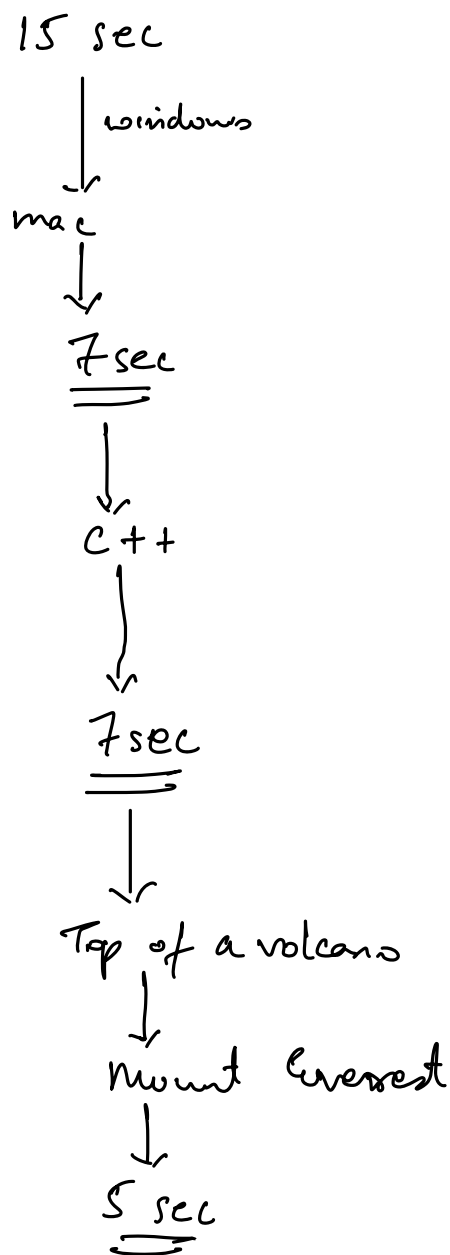
N → total numbers.

Sum of first N terms of G.P

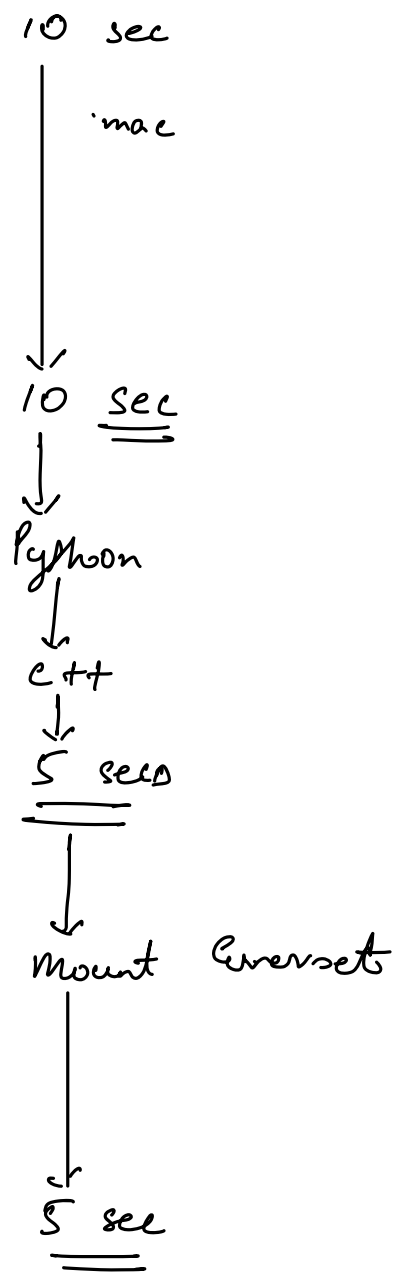
$$\text{Sum} = \frac{a (r^n - 1)}{r - 1}$$

Compare 2 algo.

Ekanth



Pallavi



⇒ Execution time cannot be deciding factor to compare algo.

Number of iterations

Next class

T.C
Big O
Space Complexity
TLE