

Introduction to Problem Solving

"Don't let what you cannot do interfere with what you can do."
— John Wooden

Hello everyone !!

Class will start at 7:10 am I.S.T

About me :

- Sakshi Sachdeva
- Senior Engineer at ZF-TCI , ex- Bosch
- B. Tech from NIT Jalandhar , 2018
- Remote Instructor at Scaler
- Born in Patiala , Punjab
- Currently residing in Hyderabad.

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.

Content for Intermediate module

- 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]

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 organised 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).
 - It'll consist of 3 questions and we expect you to solve $>=2$ problems. If for any reason you are unable to solve, then we shall also be having re-attempts as well.(We'll provide more info on re-attempts moving forward)
 - Contests are critical to retaining what you have learnt and measuring where you need improvement. Please take contests seriously.
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.
- If asking a question, ask in public chat.
- If answering a question, answer in private chat.

Lecture Days :

Monday
Wednesday
Friday



7:00 AM IST

we start at sharp 7:05

7:05

8:30

9:30

[5-10 min break]

Doubts

Double

Conceptual

- Sakshi

Implementation

- Peers
- TA

Agenda :

1. Count the factors
2. Optimisation for counting the factors
3. Check if a number is Prime
4. Sum of N natural numbers.
5. Definition of AP & GP
6. Finding no of iterations
7. How to compare algorithms

what is a factor?

i is a factor of N if i divides N completely.

How to programmatically check if i is a factor of N?

$$N \mid i \Rightarrow 0$$

Q Given N ($N > 0$), we have to count factors of N.

$$N = 24$$

Factors $\rightarrow 1, 2, 3, 4, 6, 8, 12, 24$ ans : 8

$$N = 10$$

Factors $\rightarrow 1, 2, 5, 10$ ans : 4

Idea :

min factor of a no : 1

max " " " " : N

Pseudocode :

```

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

```

N iterations

Processor : 1 GHz

Assumption : 10^8 iterations are executed in 1 sec.

N	iterations (N)	execution time
10^8	10^8	1 s
10^9	10^9	$10 s$
10^{18}	10^{18}	$10^{10} s = 317 \text{ years}$

$$10^8 \rightarrow 1 \text{ s}$$

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

$$10^9 - \frac{1}{10^8} \times 10^9 = 10$$

Optimization: $i * j = N \Rightarrow i$ & j are factors of N .
 $j = N/i \Rightarrow i$ & N/i are factors of N .

<u>$N = 24$</u>		<u>$N = 100$</u>	
<u>i</u>	<u>N/i</u>	<u>i</u>	<u>N/i</u>
1	≤ 24	1	≤ 100
2	≤ 12	2	≤ 50
3	≤ 8	4	≤ 25
4	≤ 6	5	≤ 20
6	4	10	≤ 10
8	3	20	5
12	2	25	4
24	1	50	2
		100	1

Observation: Factors are repeating | Factors exist in pairs.

- Iterate till $i \leq N/i$
 $\Rightarrow i^2 \leq N$
 $\Rightarrow i \leq \sqrt{N}$

int countFactors(int N) {

 int count = 0;

 for (int i = 1; i ≤ √N; i++) {

 if (N % i == 0) {

 if (i == N/i) {

 count = count + 1;

 } else {

 count = count + 2;

 }

 }

 return count;

}

1 : √N

√N iterations

i ≤ √N

i * i ≤ N

N = 100

count = 0 ↗ ↗ ↗ ↗ 10

i : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

N

iterations (\sqrt{N})

execution time

10^{18}

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

10 Sec

$10^8 \rightarrow 1 \text{ sec}$

$10^9 \rightarrow \frac{1}{10^8} \times 10^9 = 10$

Prime Numbers

What is a Prime number?

number which has only 2 factors.

1, N

Ques 3 : How many prime numbers are there?

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

ans: 4

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

```
boolean checkPrime (int N) {
```

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

```
        return true ;
```

```
    else
```

```
        return false ;
```

?
y

11

count = 0 2

i : 1 - - - $\sqrt{11}$ \Rightarrow 1 - - - 3 \Rightarrow 1, 2, 3

$$1, 11/1 = 1, 11$$

Sum of first N numbers

Gauss

$$\text{Quiz 4: } 1 + 2 + 3 + 4 + 5 + \dots - - - - - 99 + 100$$

$$S = 1 + 2 + 3 + 4 + 5 + \dots - - - - - 99 + 100$$

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

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

$$2S = 100 * 101$$

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

Sum of 1st N natural no

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

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

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

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

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

Some Basic math Properties

$[a, b]$: Both a & b inclusive

(a, b) : Both a & b exclusive.

Quiz 5 : How many numbers are in range $[3, 10]$

$3, 4, 5, 6, 7, 8, 9, 10$

ans : 8

Quiz 6 : How many numbers are in range $[a, b]$

$$\underline{b - a + 1}$$

$$a = 2$$

$$b = 8$$

$2, 3, 4, 5, 6, 7, 8$

$$8 - 2 + 1 = 7$$

$(a, b]$: a excluded b included

$$\begin{aligned} & (b-a+1)-1 \\ &= b-a \end{aligned}$$

$(a, b) : a, b$ excluded : $(b-a+1)-2$

$$= b-a-1$$

Break till 8:35

What is a iteration?

The number of times a loop runs, is known as iteration.

Quiz 7 : How many times will the below loop run?

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

$$1 - N \Rightarrow [1, N]$$

$$\begin{aligned} b-a+1 \\ N-1+1=N \end{aligned}$$

Quiz 8 : How many iterations will be there in this loop?

```
for(int i=0 ; i <= 100 ; i++) {  
    s = s + i + i2;  
}
```

$$\begin{aligned} [0 \rightarrow 100] &= b-a+1 \\ &= 100-0+1 = 101 \end{aligned}$$

Quiz 9 : How many iterations will be there in this loop?

func () {

for (int i = 1 ; i <= N ; i++) {

if (i % 2 == 0) {

print(i);

}

}

[1, N]

b - a + 1

N - 1 + 1

= N iteration

for (int j = 1 ; j <= M ; j++) {

if (j % 2 == 0) {

print(j);

}

}

[1, M]

M iterations

ans : N + M iterations

Geometric Progression (G.P.)

5, 10, 20, 40, 80 . . .

$$\frac{10}{5} = 2$$

$$\begin{aligned} r &= 2 \\ a &= 5 \end{aligned}$$

$$\frac{20}{10} = 2$$

$$\frac{40}{20} = 2$$

$$\frac{80}{40} = 2$$

a, ar, ar^2, \dots

a : first term of GP

r : common ratio

Sum of first N terms of a GP :

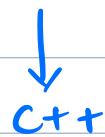
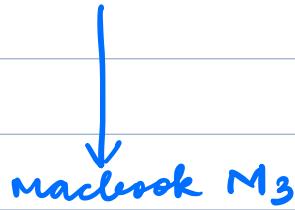
$$\frac{a(r^n - 1)}{r - 1} \quad r \neq 1$$

How to compare two algorithms?

Himanshu



(Windows XP)

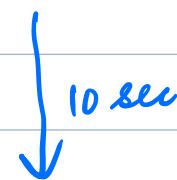


↓
near a volcano

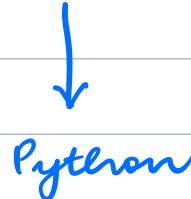
↓
mount everest



Srinivasan



(Macbook M3)



↓
5 sec
(Mount Everest)

OS, language, place of execution

Criteria to compare 2 algorithms is ITERATIONS.

Next class topics

- Big O
- Logarithm
- TLE (Time limit exceeded)
- Importance of constraints

