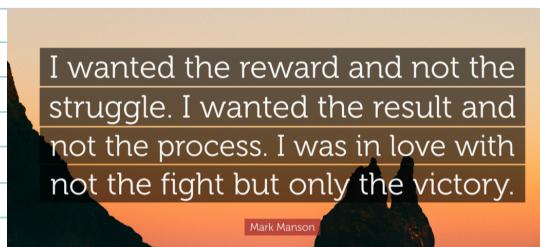


Introduction To Problem Solving



IMPORTANCE OF PSP

- Ideal ($> 100\%$)
- Red Flag ($< 80\%$)

IMPORTANCE OF ATTENDANCE

- Red Flag ($< 75\%$)

→ Note: Drop me a message why you are not able to attend.

Revision Format

- Just after the class [no matter when is class is ending]
- At the weekend
- After a month

↳ People having good PSP & Attendance Also exercises in the format. Then user will 0.1%

of Sealer students.

CLASSES IN 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]

* This module is to make us comfortable with Coding. Refresher's is for making comfortable with Java.

TODAY's CONTENT

- 1-> Count the Factors
- 2-> Optimisation for Counting the Factors
- 3-> check if a no. is prime.
- 4-> Sum of N natural nos.
- 5-> Definition of AP / GP
- 6-> How to find the no. of times a piece of code will run, i.e. no. of iterations
- 7-> How to Compare two Algorithms

What is a Factor ?

→ we can say i is a factor of N if i divides N completely, i.e. the remainder is 0

OR

$$N \% i == 0$$

Ex :- 2 is a factor of 10

$$10 \% 2 == 0 \quad , \text{it is holding true}$$

Q Given N , Count factors of N .

Note :- $N > 0$

Ques 1 :- Factor of 24

→ 1, 2, 3, 4, 6, 8, 12, 24
→ Ans 8

Ques 2 :- Factors of 10

→ 1, 2, 5, 10
→ Ans = 4

OBSERVATION :-

①

(N)

smallest factor is 1

largest factor is N

② Factor of $N \rightarrow [1, N]$

PSEUDO CODE

int factors (int N) {

 int count = 0;

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

 if (N % i == 0) {

 count ++;

 }

 return count;

3

* Standard machine in 1 sec can complete 10^8 iterations
 online compiler
 Time after which you will get TLE

* Iteration of above code

N	iterations	Time
---	------------	------

10^8	10^8	1 Sec
--------	--------	-------

10^9	10^9	10 Sec
--------	--------	--------

10^{18}	10^{18}	10^{10} Sec
-----------	-----------	---------------

≈ 317 years

10^8 iter \rightarrow 1 Sec

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

10^9 iter \rightarrow $\frac{10^9}{10^8}$ Sec \approx 10 Sec

OPTIMIZATION

$i * j = N$ { i & j both are factors of N }

Ex:- ① $6 \times 3 = 18$ & 6 & 3 are factors of 18

② $7 \times 8 = 56$ & 7 & 8 are factors of 56

$$j = \frac{N}{i}$$

↳ if i & $\frac{N}{i}$ are the factor of N ↳

Ex:-

i

N/i

$N = 24$

1	24
2	12
3	8
4	6

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

$$\rightarrow i^2 \leq N$$

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

6	4
8	3
12	2
24	1

factor
are
repeating

Ex:-

i

N/i

$N = 100$

1	100
2	50
4	25
5	20
10	10

20	5
25	4
50	2
100	1

Hence, we can
say to find a
factor N , we just
need to iterate
till \sqrt{N}

To count the factors

Ex :- : 24 \rightarrow from 1 to 4, we will get all the factors.

i	N/i	Count
1	24	+ 2
2	12	+ 2
3	8	+ 2
4	6	+ 2
		<u>8</u>

Ex :- N = 40 \rightarrow [ⁱ 1 \rightarrow 6]

i	N/i	Count
1	40	+ 2
2	20	+ 2
4	10	+ 2
5	8	+ 2
		<u>8</u>

PSEUDO CODE

```
int count factor (int N) {
    int count = 0;
    for (i = 1 ; i * i <= N ; i++)
        if (N % i == 0)
            count += 2;
    return count;
```

3

* We will get wrong ans for 100

Dry Run:-

$$N = 100$$

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

We should
not count this

10

PSEUDO CODE

```
int countFactor (int N) {
    int count = 0;
    for (i=1 ; i*i <= N ; i++) {
        if (i*i == N) {
            count++;
        } else if (N % i == 0) {
            count += 2;
        }
    }
    return count;
```

3

IMPROVEMENT

N	iterations	Time
10^{18}	10^9	10 sec

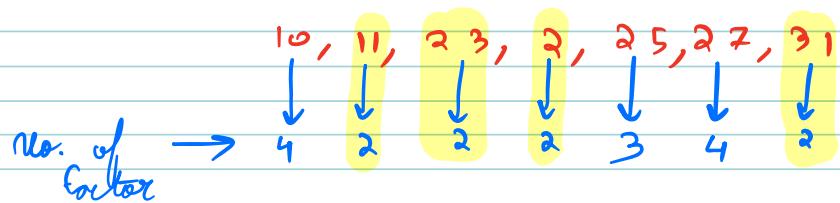
↑
improved
from 9317
years

CONCLUSION: observation is a key skill.

Q Give N, check if its prime or not.

Prime \Rightarrow Nos. with exactly 2 factors.

QUIZ 3 :- How many prime nos. are there



PSEUDO CODE :- function isPrime(int N) {

```
if (CountFactor(N)==2){  
    return true;  
} else {  
    return False;  
}
```

3

Ques 4 =>

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

$$\text{or } S = 100 + 99 + 98 + \dots + 2 + 1$$

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



100 terms

$$\therefore 2S = 100 * 101$$

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

Generalization :- Sum of $1^{\text{st}} n$ natural no.

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

$$\text{or } S = n + (n-1) + (n-2) + \dots + 2 + 1$$

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



n terms

$$\therefore 2S = n * (n+1)$$

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

← General formula

Some Basics Maths

① $[a, b]$ → This means a & b is included in the range.

② (a, b) → This means a & b is not included in the range

Ques 5 :- $[3, 10] \rightarrow 3, 4, 5, 6, 7, 8, 9, 10$
 $\underbrace{3, 4, 5, 6, 7, 8, 9, 10}_{Ans = 8 \text{ elements}}$

Ques 6 :-



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

(Total no. of
elements from
 a to b)

Iterations :- The no. of times loop is running

Ques 7 → How many times will loop run?

for ($i = 1$; $i \leq N$; $i++$) {

 |
 if ($i == N$) break;

 3

This loop is running from $[1, N]$

∴ Total iterations = $N - 1 + 1 = N$ Ans

Quiz 8 :- How many iterations will be there in loop.

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

$$\text{Total Iterations} = [0 \ 100]$$

$$= b-a+1$$

$$= 100-0+1$$

$$= 101 \quad \underline{\text{Ans}}$$

Quiz 9 :- Count Total Iterations

```
fun() {
```

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

```
        if ( i%2 == 0 ) {
```

```
            Print(i);
```

[1 N]

$\Rightarrow N-1+1$

$\Rightarrow N$

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

```
        if ( j%2 == 0 ) {
```

```
            Print(j);
```

[1 M]

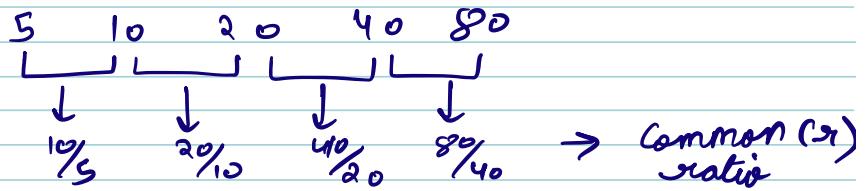
$\Rightarrow M-1+1$

$\Rightarrow M$

since the two loops are parallel

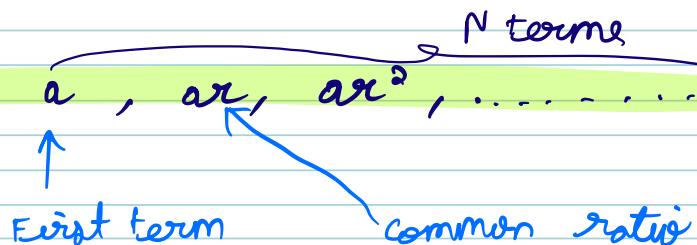
$$\therefore \text{Total Iterations} = N+M:$$

Geometric progression [G.P.]



* In GP 'r' remains same

Generic Notations



$$\Rightarrow \text{Sum of GP} = \frac{a[r^n - 1]}{r - 1} \quad [r \neq 1]$$

Ex:- 5, 10, 20, 40, 80

$$\text{here } a = 5, r = \frac{10}{5} = 2$$

$$n = 5$$

$$\begin{aligned} \text{Sum} &= \frac{a(r^n - 1)}{r - 1} = \frac{5(2^5 - 1)}{2 - 1} \\ &= 5(32 - 1) = 155 \end{aligned}$$

Comparing Two Codes

Let us compare the sorting code

Eg:- arr = 2 5 4 1 3 3

= 1 2 3 3 4 5

SAI

↓
15 Sec

↓
Window xp

↓
M2

↓
7 Sec

↓
C++

↓
7 Sec

↓
Top of Hot Volcano

↓
Mount Everest

↓
5 Sec

EKTA

↓
10 Sec

↓
M2

↓
10 Sec

↓
Python

↓
C++

↓
5 Sec

↓
5 Sec

CONCLUSION

We can't compare two algo on the basis of their execution time as it depends of factors like OS, place, language etc.

Questions

- How to compare?
- which doesn't depend on any factor.

Answer \Rightarrow No. of iterations

[The no. of iterations won't depend on factors like OS, language, place etc.]

Next class Content

- Big O Notation
- logarithms
- space complexity
- TLE & Importance of constraints

DOUBT SESSION

(1)

math. Pow (long, 3)

math. Pow (int, 3)

(2)

$$\begin{array}{r}
 0000 \\
 \underline{\quad\quad\quad} \\
 0024 \\
 \underline{\quad\quad\quad} \\
 0036
 \end{array}
 \quad
 \begin{array}{r}
 5 \\
 \underline{\quad\quad\quad} \\
 8 \\
 \underline{\quad\quad\quad} \\
 9
 \end{array}
 \quad
 \begin{array}{r}
 0000 \\
 \underline{\quad\quad\quad} \\
 1200 \\
 \underline{\quad\quad\quad} \\
 12
 \end{array}
 \quad
 \begin{array}{r}
 0000 \\
 \underline{\quad\quad\quad} \\
 0000 \\
 \underline{\quad\quad\quad} \\
 1500
 \end{array}$$

$2 \times \text{Row-1}$



Row	STO	EDO	C Numbe
1	4	4	1
2	3	3	3
3	2	2	5
4	1	1	7
5	0	0	9