Hello Everyone 

→ Welcome to Intermediate module of D.S.A

→ Jitender Punia (Jeetu)

→ B.Tech from VSICT, co-founder of pepcoding

→ ~3 years of teaching experience.

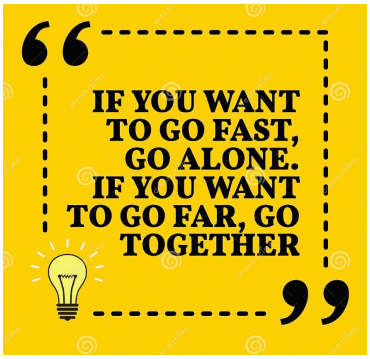
FAQ's.

→ Notes will be uploaded after the class.

→ Assignments will be unlocked after the class ends.

→ There is no deadline for assigments.

Today's Quote →



→ Peer-to-peer learning

Q: Count of factors.

⇓

any no. which divides N completely.

`N % i == 0`

N = 24. { 1, 2, 3, 4, 6, 8, 12, 24 }.

N = 10 { 1, 2, 5, 10 }

[N % i == 0] ⇒ i is a factor of N.

pseudo-code

```
int countFactors (int N) {
    int factors = 0;
    for( int i = 1; i <= N; i++){
        // is i a factor of N
        if ( N % i == 0){
            factors = factors + 1;
        }
    }
    return factors;
}
```

→ N ?
→ system ?

i → [1 to N]

⇓

{N iterations}

{Assumption. → 10^8 iterations per sec}

N	Iterations	Execution time?
10^8	10^8	1 sec.
10^9	10^9	10 sec.
10^{18}	10^{18}	10^{10} sec. ≈ 317 years.

You → children → grand-children → 4th → 5th / 6th

Optimise

$i * j = N$ ⇒ { i and j are factors of N }

⇒ $\boxed{j = N/i}$ { i and $\frac{N}{i}$ are factors of N }

N = 24

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

N = 100

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

observations → After a particular no, factors are repeating.

→ All the factors are present in part 1.

→ In part-1 → $i <= \frac{N}{i}$

$i * i <= N$

$i^2 <= N$ ⇒ $\boxed{i <= \sqrt{N}}$

```
int countFactors(N){
    factors = 0;
    for(i = 1; i <= √N; i++){
        if(N%i == 0){
            if(i == N/i) factors++;
            else factors = factors + 2;
        }
    }
    return factors;
}
```

$i : [1, \sqrt{n}]$

$\sqrt{n}$ iterations

// i and N/i are two factors

N = 25

$N = 10^8$	no. of iterations = 10^9	execution time = 10 sec.

⇒ Most important skill for problem solving → observation

Q: Given N. You need to check if this is prime or not.

Nos → {10, 11, 23, 2, 25, 27, 31}

prime → no. which is only having 2 factors.
⇓
[1 & N]

```
boolean checkPrime (N){
    if (countfactors(N) == 2) {
        return true;
    }
    else {
        return false;
    }
}
```

Game :

4th class. // Gauss

$$S = 1 + 2 + 3 + 4 + \cdots\cdots 98 + 99 + 100$$

$$+\ S = 100 + 99 + 98 + 97 + \cdots\cdots 3 + 2 + 1$$

$$2S = 101 + 101 + 101 + 101 + \cdots\cdots 101 + 101 + 101$$

$$2S = (101)(100)$$

$$S = \frac{(100)(101)}{2}$$

Sum of 1st _ N natural no's.

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

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

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

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

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

Q.1 Given N → perfect square. Find sqrt(N).

N = 25 → 5

N = 36 → 6

N = 49 → 7

N = 30 {We will never get invalid inputs}

```
int sqrt(N) {
    for(int i=1; i<=N; i++) {
        if(i*i == N) return i;
    }
}
```

Amazon MCQ.

(a) $\log_2 n$

(b) N

(c) ~~(c)~~ $\sqrt{N}$

(d) N.O.T.

N = 25, i = 1, 2, 3, 4, 5

Q.1 Sqrt(N)

Note → If N is not a perfect square → floor(sqrt(N)).

N = 49 → 7

N = 60 → 7

N = 31 → 5

N = 29 → 5

[floor(x) → Greatest Integer value <= x]

N = 50.

i	i*i <= N
1	ans = 1
2	ans = 2
3	ans = 3
4	ans = 4
5	ans = 5
6	ans = 6
7	ans = 7
8	8*8 <= 50 ✗

```
int sqrt(N){
  int i=1, ans=0;
  while(i*i <= N){
      ans = i;
      i++;
  }
  return ans;
}
```

No. of iterations ⇒ $\sqrt{n}$

N = 36 ⇒

i = 1	ans = 1
i = 2	ans = 2
i = 3	ans = 3
i = 4	ans = 4
i = 5	ans = 5 ✓
i = 6	ans = 6
i = 7	7*7 > 36 break

sqrt(N)
- idea-1 → $\sqrt{n}$ iterations
- idea-2 → $\log_2 n$ iterations [Advanced - module]

Log - Basics

$$\log_b a = c$$

[to what value we need to raise b to get value = a]

$$\Rightarrow b^c = a$$

$\log_2 64 = 6$

$\log_3 27 = 3$

$\log_5 25 = 2$

$\log_2 32 = 5$

$\log_2 10 = 3$ ✓

$\log_2 40 = 5$

$N = 2^K$ (given), $K = \log_2 N$ ***

$\log_2 2^{10} = 10$

$\log_3 3^5 = 5$

$\Rightarrow \log_a a^n = n$ ***

Q.1 Given +ve integer N. How many times we need to divide it by 2 until it reaches 1.

N = 100

/2 → 50

/2 → 25

/2 → 12

/2 → 6

/2 → 3

/2 → 1

ans = 6

N = 324

/2 → 162

/2 → 81

/2 → 40

/2 → 20

/2 → 10

/2 → 5

/2 → 2

/2 → 1

[Homework].

ans = 8

N = 1024

⇩

ans = 10

Intermediate content

- Introduction to Problem Solving ✓
- Time Complexity - 1, 2
- Arrays - 6 {prefix Sum, subarrays, carry-forward, sliding window, 2-D matrix}
- Interview Problems - 2
- Bit manipulations - 3
- Modular Arithmetic - 1
- Sorting - 1 ✓
- Strings - 1 ✓
- Hashing - 2 ✓
- Recursion - 2 ✓
- [Classes & objects - 1]
- {Linkedlist Basics - 1
- Trees Basics - 1}

2 months.

{Jun → problem solving}.

$\log_2 2^{10} = 10.$

200 Questions.

$\log_2 3^{10} =$

$\log_3 3^{10} \rightarrow 10$

$\log_2 1024$

$2^{?10} = 1024$

$\log_2 2^{10}$