Hello Everyone 

→ Welcome to Intermediate module of D.S.A

→ Jitender Punia (Jeetu)

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

→ ~3.5 years of teaching experience.

FAQ's.

→ Notes will be uploaded after every class.

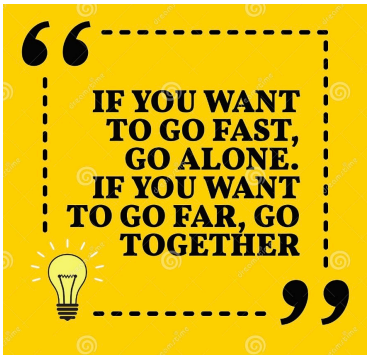
→ Assignments will be unlocked after the class ends.

→ There is no deadline on assignment.

answer: To: Jitender Punia.

question: To: Everyone
or
Questions stack.

Today's Quote →




[peer to peer learning]

# Count of factors

24 → [1, 2, 3, 4, 6, 8, 12, 24] ans = 8.

10 → [1, 2, 5, 10] ans = 4.

# pseudo-code

[N? System?]

```
int countOfFactors(int N){
    int factors = 0;
    for(int i = 1; i <= N; i++){ // i : [1,N] => N iterations.
        if(N % i == 0){ // i is a factor of N
            factors += 1;
        }
    }
    return factors;
}
```

//Assumption - $10^8$ iterations per sec.

| N | iterations | Execution Time? |
|---|---|---|
| $10^8$ | $10^8$ iterations | 1 sec. |
| $10^9$ | $10^9$ iterations | 10 sec. |
| $10^{18}$ | $10^{18}$ iterations | $10^{10}$ sec ≈ 317 years |

$10^8$ iterations $\rightarrow$ 1 sec

1 iteration $\rightarrow \frac{1}{10^8}$ sec.

$10^9$ iterations $\rightarrow \frac{1}{10^8} \times 10^9$ sec.

$= 10$ sec.

**observation**

$i * j = N$ [Both $i$ and $j$ are factors of $N$]

$j = N/i$ [Both $i$ and $N/i$ 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 |

obs. - After a certain value, factors are repeating.

All the factors are present in part-1.

In part-1.

$$i \leq N/i$$

$$i * i \leq N$$

$$i^2 \leq N \Rightarrow \boxed{i \leq \sqrt{N}}$$

```
int countofFactors(int N){
    int factors = 0;
    for(int i = 1; i*i <= N; i++){   // i : [1,√N] => √N iterations.
        if(N%i == 0){ // i, N/i are factors
            if(i == N/i) { factors += 1 }
            else { factors += 2 }
        }
    return factors;
}
```

N = 100

factors = 0 2 4 6 8 10

i : 1 2 3 4 5 6 7 8 9 10 11

//Assm.. $10^8$ iterations per sec.

| N | iterations | Execution time? |
|---|---|---|
| $10^{18}$ | $10^9$ iterations | 10 sec. |

[Most important skill for problem solving → Observation]

## Prime Number

Number having only 2 factors.

[1 and N itself]

Q: How many prime numbers are there?

[10, 11, 23, 2, 25, 27, 31] ans = 4.

Q. Given N. Check whether it is prime or not.

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

4th class. // Gauss

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

$S = 100 + 99 + 98 + 97 + 96 + \dots + 2 + 1$

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

$2S = 101 * 100$

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

Sum of first N natural no's

$S = 1 + 2 + 3 + 4 + 5 + \dots + (N-1) + N$

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

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

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

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

Q. 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(i=1; i<=N; i++){
        if(i*i == N){
            return i;
        }
    }
}
```

Amazon MCQ

(a) $\log_2 N$

(b) N

(c) $\sqrt{N}$

(d) None of these.

N = 36. i = ~~1 2 3~~ 4 ~~5~~ 6

N = 49. 7

N = 64 8

Q: Find sqrt(N)

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

N = 49 ⇒ 7

N = 60 ⇒ 7

N = 31 ⇒ 5

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){
    i = 1, ans = 1
    while(i*i ≤ N){
        ans = i;
        i++;
    }
    return ans;
}
```

# no. of iterations → $\sqrt{N}$

sqrt(N) → $\log_2 N$ iterations

[Advanced module]

# Log Basics

$\log_b a = c$ ⇒ to what power $b$ must be raised to get value $a$.

$b^c = a$

$\log_2 64 = 6$ $[2^6 = 64]$

$\log_3 27 = 3$ $[3^3 = 27]$

$\log_5 25 = 2$ $[5^2 = 25]$

$\log_2 32 = 5$ $[2^5 = 32]$

$\log_2 2^{10} = 10$ $[2^? = 2^{10}]$

$\log_3 3^5 = 5$ $[3^? = 3^5]$

$\log_a a^N = N$

$[N = 2^K]$

$\log_2 N = \log_2 2^K$

$[\log_2 N = K]$

$\log_2 10$ — $2^3 = 8$, $2^4 = 16$ — $3.x = 3$

$\log_2 40 = 5$

Q: 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 = 15

↓ /2

7

↓ /2

3

↓ /2

1

[ans = 3]

#code:-

```
N → given
count = 0
while ( N > 1 ) {
        N = N/2
        count ++ ;
}
return count;
```

# no of iterations → [H.W]

$i \leq \sqrt{N}$

$i^2 \leq N$

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

$\sqrt{N}$ iterations.

T.C → 2

Arrays → 5 - 6 sessions

B.M →

Maths → modular arithmetic

Sorting

String

Hashing · 2

Recursion

L·L

150 - 180 problems.

Arrays

B.M

Maths → P.C
→ modular arithmetic
→ Prime n.
→ G.C.D

Recursion

Sorting

Searching → Binary Search

Hashing

Stack

Queues

Linked list

Trees + Trie

Heap

Backtracking

D.P

Graphs.

→ N

→ $\sqrt{N}$

→ $\log_2 N$