# Welcome

## Intermediate module

### FAQs

1. Notes will be uploaded
2. Assigments
3. Homework
4. PSP → Problem solving percentage.
   > 90%
5. Contest
   3 contest in intermediate
6. How to solve a problem.
   1. Pen and paper → 30 minutes
      Then implement on IDE
   2.

questions ⇒ public chat
answers ⇒ private chat

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

factors ⇒ any number i which divides N completely ∴ a factor

$N \% i == 0$

24 { 1, 2, 3, 4, 6, 8, 12, 24 } ⇒ 8

10 { 1, 2, 5, 10 } ⇒ 4

Pseudo code

```
int countfactors ( int N) {
    int factors = 0 ;
    for( i=1 ;  i ≤ N ;  i++)
    {
        if ( N % i == 0)
            factors += 1;
    }
    return factors ;
}
```

1. N
2. system

Assumption : $10^8$ iterations ≃ 1 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 yrs |

# Optimization

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

$j = \frac{N}{i}$ { $i$, $\frac{N}{i}$ both are factors }

$N = 24$

| $i$ | | $N/i$ |
|---|---|---|
| 1 | $\leq$ | 24 |
| 2 | $\leq$ | 12 |
| 3 | $\leq$ | 8 |
| 4 | $\leq$ | 6 |
| 6 | $\geq$ | 4 |
| 8 | $\geq$ | 3 |
| 12 | $\geq$ | 2 |
| 24 | $\geq$ | 1 |

$P_1$

$N = 100$

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

$P_1$

$P_2$

Obs 1: After certain no., factors are repeating.

Obs 2: In part 1, $i \leq \frac{N}{i}$

$i^2 \leq N$

$i \leq \sqrt{N}$

Optimi code

N = 100

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

| i = 1 | fa |
|---|---|
| ↓ | 2 |
| 2 | ↓ |
| ↓ | 4 |
| 3 | ↓ |
| ↓ | 6 |
| 4 | |
| ↓ | |
| $\sqrt{N}$ | |

$\sqrt{N}$ iterations

| N | iterations | execution time |
|---|---|---|
| $10^{18}$ | $10^9$ | 10 sec |

---

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

⇒ { 10, 11, 23, 2, 25, 27, 31 } (4)

(10 ✗, 11 ✓, 23 ✓, 2 ✓, 25 ✗, 27 ✗, 31 ✓)

prime numbers have only 2 factors → 1 & N

Pseudo code

```
bool checkPrime (int N)
{   if( countfactors (N) == 2)
        return true
    else
        return false
}
```

Gauss

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

$$S = 100 + 99 + \ldots + 2 + 1$$

$$2S = 101 + 101 + \ldots + 101$$

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

Sum for 1st N natural numbers.

$$S = 1 + 2 \ldots + N$$

$$\text{Sum} = \frac{N * (N+1)}{2}$$

---

Q Given a no. N → perfect square. Find sqrt N.

eg: 25 ⇒ 5
36 ⇒ 6
100 ⇒ 10
30 ⇒ ✗

pseudo code

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

N iterations

Q. Find sqrt N, given that N is not a perfect square.. Return floor( sqrt(N))

N = 49 ⇒ 7
60 ⇒ 7
31 ⇒ 5

floor(n) ⇒ floor(3.2) = 3
floor(3.98) = 3

Pseudo code

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

N = 50

| i | i*i ≤ N |
|---|---|
| 1 | ans = 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | |

# iterations ⇒ i → [1, √N]

√N iterations.

# Log - Basic

$\log_a b = c \Rightarrow$ to what values we need to raise 'a' so that the value becomes 'b'.

$a^c = b$

$\log_2 64 \Rightarrow 2^? = 64 = 6$

$\log_3 27 \Rightarrow 3^? = 27 = 3$

$\log_5 25 \Rightarrow 2$

$\log_2 32 \Rightarrow 5$

$\log_2 10 = 3$

$\log_2 40 = 5$

$\Rightarrow \log_2 2^{10} \rightarrow 10$

$\log_3 3^5 \rightarrow 5$

$2^K = N$

$K = \log_2 N$

$\log_a a^m = m$

Q Given a +ve integer N. How many times do we need to divide it by 2 untill it becomes 1.

N = 100
↓ /2
50
↓ /2
25
↓ /2
12
↓
6
↓
3
↓
1

Ans = 6

HW

## Intermediate module

1. Intro
2. Time Complexity → (2)
3. Arrays → (6)
4. interview problems
5. Contest 1 *
6. Bit manipulation. (2)

7. Modular arithmetic
8. Sorting.
9. Interview problems.
10. Contest 2 ☆
11. String
12. Hashing (2)
13. Recursion (2)
14. Linked lists
15. Tree basics
16. Contest 3 ☆