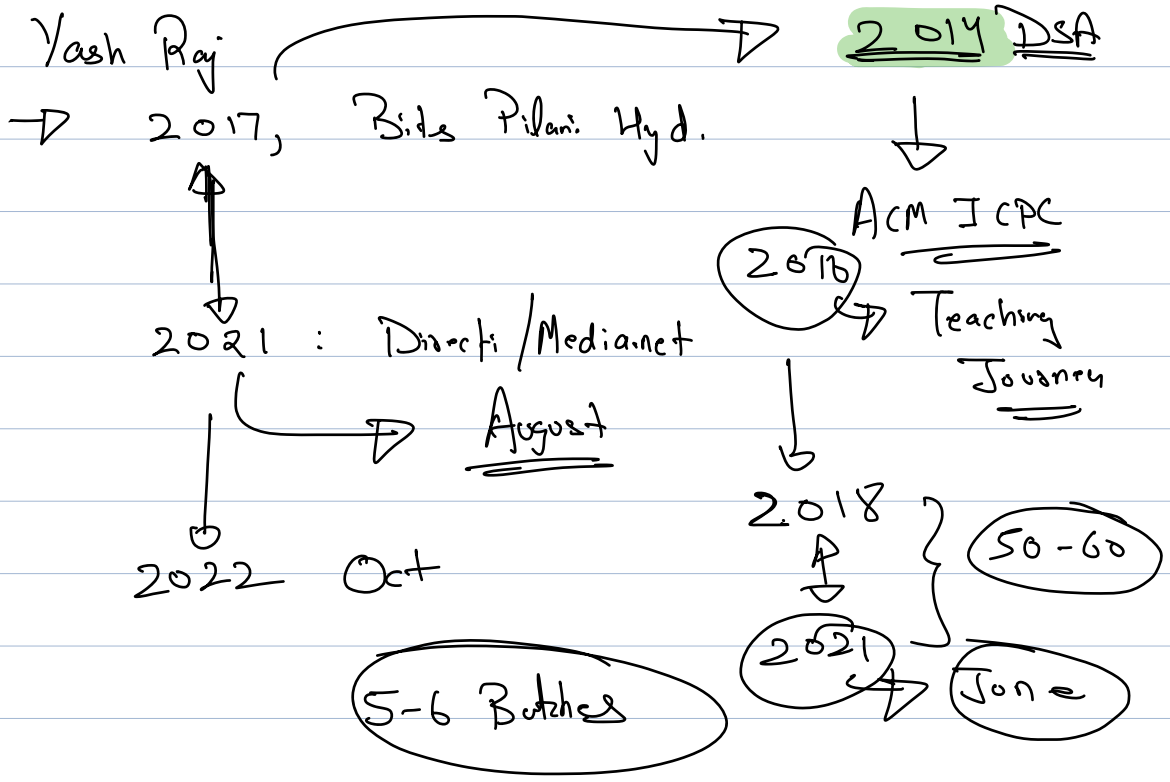

- 2:30 hours long.
- DSA → Pseudo Codes

- Lectures ⇒ 1
- Assignment ⇒ 2
- Homework ⇒ 3

⇒ 24 hours
Sleep = 8
office ⇒ 8
food | bath | exercise
⇒ 3
⇒ 5 - 3
⇒ 2.

5

Count of factors

↓

any number which divides N completely.

N % i == 0

↳ i is a factor of N

N = 24 : Count of factors ⇒ {1, 2, 3, 4, 6, 8, 12, 24}

↳ 8

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

↳ 4.

Pseudo Code

```
int countfactors (int N) {
    int ans = 0
    for (int i = 1; i <= n; i++) {
        if (N % i == 0)
            ans++;
    }
    return ans;
}
```

Assumption

10^8 iteration $\Rightarrow$ 1 sec

N	iteration	Execution time.
10^8	10^8	1 sec
10^{10}	10^{10}	100 sec
10^{18}	10^{18}	10^{10} sec
		$\Downarrow$
		317 years

10^8 iteration $\Rightarrow$ 1 sec

1 iteration $\Rightarrow$ $\frac{1}{10^8}$ sec

10^{10} iteration $\Rightarrow$ $\frac{1}{10^8} \times 10^{10} \Rightarrow 10^2$

Brute force $\Rightarrow$ Worst solution possible.

⇒ N if i is a factor.

$i \times j = N$

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

Observation 1: factors come in pairs.

$N = 24$

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

$N \Rightarrow 100$

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

$i \leq N/i$ ⇒ $i \times i \leq N$

$i^2 \leq N$

$i \leq \sqrt{N}$

Pseudo Code

$i \leq \sqrt{n}$

$x = sqrt(n)$

$i \leq x$

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

Total iterations $= \sqrt{N}$

$N = 36$

i	ans	N/i
1	2	(36)
2	4	(18)
3	6	(12)
4	8	(9)
5	8	
6	9	(6)
7		

ans = 9

N	iteration	Execution time.
10^{18}	10^{9}	10 seconds

OBSERVATION ⇒ Most important skill.

8:50 am

Q2 Given N. Check if N is a prime number.

↓

count of factors = 2

Prime numbers are numbers which have factors 1 and the number itself only.

N = 1 except 1

Pseudo Code

```
bool checkPrime (int N) {

    int cf = countFactors (N);
    if (cf == 2)
        return true;
    else
        return false;

}
```

H.W : Optimised version

$S = 1 + 2 + 3 + 4 \quad \ldots\ldots \quad + 100$

$S = 100 + 99 + 98 + 97 \quad \ldots\ldots \quad + 1$

$2S \Rightarrow 101 + 101 + 101 + 101 \quad \ldots\ldots \quad 101$

$2S \Rightarrow 100 \times 101$

$S \Rightarrow \frac{100 \times 101}{2} \Rightarrow 5050$

$\Rightarrow$ GUASS

$S \Rightarrow 1 + 2 + 3 \ldots\ldots\ldots\ldots n$

$S \Rightarrow n + (n-1) + (n-2) \ldots\ldots\ldots\ldots 1$

$2S \Rightarrow (n+1) + (n+1) + (n+1) \ldots\ldots\ldots\ldots (n+1)$

$2S \Rightarrow n \times (n+1)$

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

Q4) Given a perfect Square number N.
Find the square root of N

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

Amazon MCQ

1) $\log_2 N$
2) N
3) $\sqrt{N}$
4) None of these.

Qs Find sqrt(N)

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

floor(x) → Greatest integer ≤ x

floor(7.2) → 7
floor(2.9) → 2
floor(3) → 3

i	i×i	ans
1	1	1
2	4	2
3	9	3
4	16	4
5	25	5
6	36	6
7	49	7
8	64	break

N = 50

22 – 26