Prime no.s → Only 2 factors (1 & itself)

Eg - 5, 11, 17

→ count_of_factors → == 2 is prime
→ ! = 2 not prime

- How to write is_prime function ?

Iterate till $\sqrt{N}$ & get count of factors

Q1 Given N, find all primes from 1 to N

N=10 → { 2 3 5 7 }

N=20 → { 2 3 5 7 11 13 17 19 }

Brute force : Iterate on 1 to N & check if prime or not for each number

```
for (i=1 ; i ≤ N ; i++)
        if (is_prime (i)) {
            print (i)
        }
}
```

TC: $O(N\sqrt{N})$

SC: $O(1)$

Given N= 30, find all primes

1 2 3 4 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30

Seive of erastos thenes

$N+1 \rightarrow N$

$N \rightarrow N-1$

Code

```
bool p[N+1] = { True }    // all val = true
p[0] = p[1] = false
for (i=2 ; i≤n ; i++) {
    if (p[i] == true) {
        for (j=2i ; j≤n ; j+=i) {
            p[j] = false
        }
    }
}
```

Now wherever p[i] = true,
i is prime

TC:

i	j
2	$n/2$
3	$n/3$
~~4~~	
5	$n/5$
N	

Total = $N/2 + N/3 + N/5 + N/7 + \cdots$

$$= N\left[\underbrace{\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \cdots}_{\log(\log(n))} \right] \qquad \frac{1}{\text{prime}}$$

TC: $O(N \log\log(N))$

SC: $O(N)$

Q2 Given N, find smallest prime factor for all no.s 2 to N.

Eg
10 → 2
15 → 3
17 → 17
35 → 5

N = 10

	2	3	4	5	6	7	8	9	10
⇒	2	3	2	5	2	7	2	3	2

N = 30

1	2	3	4	5	6	7	8	9	10
1	2	3	2	5	2	7	2	3	2
11	12	13	14	15	16	17	18	19	20
11	2	13	2	3	2	17	2	19	2
21	22	23	24	25	26	27	28	29	30
3	2	23	2	5	2	3	2	29	2

SPF variation

Code

LOOP

```
spf[N+1] = {  intialize spf[i] = i }
for (i=2 ; i≤n ; i++) {
    if ( spf[i] == i ) {
        for (j=2i ; j≤n ; j+=i) {
            spf[j] = min(spf[j], i)
        }
    }
}
```

TC: $O(n \log_2(\log_2 n))$ SC: $O(N)$

$N = 10^5$

n^2

10^{10}

$10^5 \times 10^5$

$n \log(\log(n))$

$10^5 (\log(18))$

$10^5 \times 4$

break back at 10:10

Q3 Count no of divisors

Eg 1 72 $\Rightarrow 2^3 \times 3^2$

$2^0 \times 3^0 = 1$	$2^0 \times 3^1 = 3$	$2^0 \times 3^2 = 9$	$[2^0\ 2^1\ 2^2\ 2^3]$
$2^1 \times 3^0 = 2$	$2^1 \times 3^1 = 6$	$2^1 \times 3^2 = 18$	$[3^0\ 3^1\ 3^2]$
$2^2 \times 3^0 = 4$	$2^2 \times 3^1 = 12$	$2^2 \times 3^2 = 36$	4×3
$2^3 \times 3^0 = 8$	$2^3 \times 3^1 = 24$	$2^3 \times 3^2 = 72$	$(3+1)(2+1)$

Eg 2 600 $\Rightarrow 2^3 \times 3^1 \times 5^2$

ans = $(3+1)(1+1)(2+1)$

$4 \times 2 \times 3 = 24$

Generalization

$$N = p_1^{x_1} p_2^{x_2} p_3^{x_3} \cdots\cdots p_k^{x_k}$$

$p_1, p_2, \ldots p_k$ are primes

Factors = $(x_1+1)(x_2+1)(x_3+1)\cdots\cdots(x_k+1)$

$N = 360$ spf $= 2$

keep dividing by spf untill cannot continue

$360 \to 180 \to 90 \to 45$

power of $2 = 3$

multiplier $= 3+1 = 4$

spf of $45 = 3$

$45 \to 15 \to 5$

power $= 2$

mult $= 2+1 = 3$

$5 \to 1$ STOP

power $= 1$

mult $= 1+1 = 2$

Total no of factors $= 4 \times 3 \times 2$
$= 24$

$1 \times 4 \times 3 \times 2 = 24$

Code

1) Create spf array $\Big]$ TC: nloglogn

```
int get_num_of_factors (int N) {
    total = 1
    while ( N != 1 ) {
        p = spf[N]
        count = 0
        while ( N % p == 0 ) {
            count ++
            N = N/p
        }
        total = total * (count+1)
    }
    return total
}
```

TC: log(n)

TC: $\boxed{n \log \log n} + \log n$

nloglogn

Q4 Given N, for all 1-N, find no of factors

	1	2	3	4	5	6	7	8	9	10
N=10	1	2	2	3	2	4	2	4	3	4

Idea: Use get_num_of_factors

Code

1) Create spf array } $O(n \log\log(n))$

```
cnt [N+1]
for( i=1 ; i<=N ; i++) {
    cnt [i] = get_num_of_factors (i)
}
```

nlogn

TC: nlog(n)

{done}