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

Eg - 2, 3, 5, 7, 11, 13, ...

count_of_factors ==2 → is prime

count_of_factors !=2 → not prime

- How to write is_prime function ?

Iterate till sqrt(N) & get count of factors

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

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 (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

✗	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

Prime ⇒ ●

Seive of Eratosthenes

23 46 69 92

2×23 3×23 4×23

N ⇒ N-1

N+1 ⇒ N

Code

```
bool p[N+1] = {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
        }
    }
}
```

TC :

SC : $O(n)$

Now wherever $p[i]$ = true,

i is prime

TC:

i	j
2	n/2
3	n/3
5	n/5
7	⋮
⋮	
N	

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

$= N \left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \cdots \right]$

↳ $\log(\log(n))$

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 spf

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

11	12	13	14	15	16	17	18	19	20
11	~~12~~ 2	13	~~14~~ 2	~~15~~ 3	~~16~~ 2	17	~~18~~ 2	19	~~20~~ 2

21	22	23	24	25	26	27	28	29	30
~~21~~ 3	~~22~~ 2	23	~~24~~ 2	~~25~~ 5	~~26~~ 2	~~27~~ 3	~~28~~ 2	29	~~30~~ 2

arr(n+1) 0 1 2 3 ... n-1 n

Code

```
int spf[N+1];
for(i=0; i<=N; i++)
    spf[i] = i
```

```
for(i=2; i<=n; i++){
    if(i == spf[i]){
        for(j=2i; j<=n; j+=i){
            spf[j] = min(spf[j], i)
        }
    }
}
```

TC: SC: Same

$360 = 8 \times 9 \times 5$

$= 2^3 \times 3^2 \times 5^1$

Q3 Count no of factors

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

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

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

ans = $(3+1)(1+1)(2+1) = 24$

Generalization

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

$p_1, p_2, \cdots p_k$ are primes

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

N = 360 spf = 2 factors = 1

keep dividing by spf untill cannot continue

360 → 180 → 90 → 45

power of 2 = 3 factors = 4

spf of 45 = 3

45 → 15 → 5

power = 2 factors = 12

5 → 1 spf = 5 factors = 24

power = 1

$2^3 \; 3^2 \; 5^1$

135 spf = 3 $3^3 \times 5^1$

135 → 45 → 15 → 5 power = 3

5 spf = 5

5 → 1 power = 1

Code

1) Create spf array

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

```
int get_number_of_factors(int N){
    factors = 1
    while(N!=1){
        p = spf[N]
        power = 0
        while(N%p==0){
            N = N/p
            power++
        }
        factors = factors * (power+1)
    }
    return factors
}
```

TC: $\log(n)$

$n \log\log n + \log n \Rightarrow O(n \log\log n)$

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	3	3	4

Idea: Use get_no_of_factors function

Code

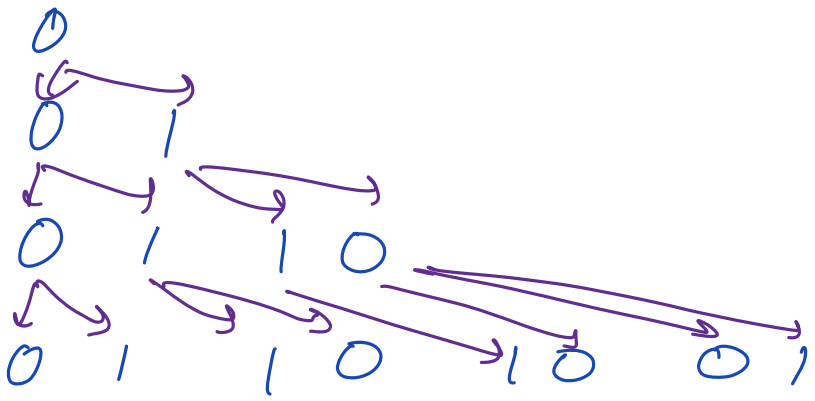
1) Create spf array
2) cnt[N+1]

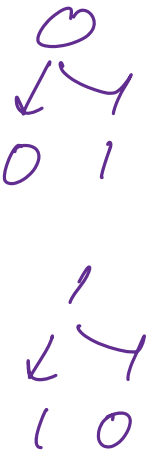
```
for(i=1; i<=N; i++){
    cnt[i] = get_no_of_factors(i)
}
```

TC: $N \log\log n + N \log n \Rightarrow O(N \log N)$

SC: $O(n)$

0 1 2 3 4 5 6 7





left child = par

right child = ~par

parent of i, j

$(i-1, j/2)$

if $j\%2 == 0$ left child

else right child

$$x \quad x+1 \quad x+2 \quad \cdots \quad x+k$$

$$x + (x+1) + x+2 \quad \cdots\cdots \quad (x+k) = A$$

$$\frac{(k+1)2x}{x} + \frac{k(k+1)}{x} = 2A$$

$$(k+1)[2x+k] = 2A$$

$$2A$$

$$k+1 = i$$

$$k = i-1 \qquad = 2x+k$$