CLASS-37

Number Theory

→ Modular Arithmetic
- → What is Division Mod
- → why 1e9+7 as Mod
- → (a+b)%M , (a-b)%M , (a*b)%M

→ GCD, LCM
- → a*b = GCD*LCM
- → Euclid's Algorithm ; __gcd( ) ;
- → factors / Divisors
- → count , sum

$N = p_1^a * p_2^b * p_3^c \ldots$

$$\begin{cases} count = (a+1)(b+1)(c+1)\ldots \\ sum = \left(\frac{p_1^{a+1}-1}{p_1-1}\right) * \left[\frac{p_2^{b+1}-1}{p_2-1}\right] \ldots \end{cases}$$

Prime factorization of a Number
- → Prime Nos , How to check
  - → O(n)
  - → O(√n)
- → How to find Prime factorization

2 → even Prime
Prime → odd

n → for (i = 2 ; i <= n ; i++)
int cnt = 0 ;
while (n%i == 0) { n = n / i ; cnt++ ; }

i = 2, i = 3 → only Prime Nos

2 - 3√n

O(n)

n = 1

n=

Only Prime No's

Optimized

magram?

Prime Nos

```
if(n==1)
  break;
}
```

Composite No

cnt++;

n=1

n=1

n=

36 = (2^2 * 3^2 * 1) 12) 12

n=1 =>

36 =>

36 = 1 => 9*x / 9

i=9

i=3

9

x / y => y*L / y

n / c => *x / c

p1, p2, p3 < c

c = p1 * p2 * p3

50 = 2 * 5 * 5

50 / 2 = 2*x / 2

50 = 5*5

for(i=2 ... i<=50; i++)

50

i=3 4 5

50 = 3x / 3

2 * 5

50 / 5 = 5x / 5 => 5*1 / 5 = 1

6, 7, 8, 9, 10

50 / 10 = 10*x / 10

2nd optimization

```
while(n%2==0)
{ n/=2;
}
```

4, 6, 8, 10, 12

```
for(i=3; i<=n; i+=2)
```

3, 5, 7

n odd

2, odd

Prime NOC

```
for (i=3; i<=n; i+=2)
{
    while (n%i==0)
    {
        n1=i;
        n
    }
}
```

3rd optimization

$i=2, \ldots, i \le n$

$i = \sqrt{n}$

> 100

$n = i^2 * i^2 * i^2 * 1$

$(i+1)$

$n = i^2$

```
for (i=3; i*i<=n; i+=2)
{
    while (n%i==0)
    {
    }
}
```

7, 11

$i = \sqrt{n}$

$(i+1)$

$i*i \le n$

$(i+1) > \sqrt{n}$

$i*i > n$

$22 = 11$

11^2

```
for (i=3; i<=4; i++)
{
}
if (n > 1)
    Prime_factor.push_back(n);
```

$\Longrightarrow$

```
for (i=3; i*i<=n; i++)
{
    while
    {
    }
}
```

$i*i \le n$; $i = p_1$; $p_1^2 \le n$; $p_1 = 3$; $9 \le 27$

$n = p_1^3$

$n = 3*3*3 = 27$

{ prime_factor.push_back(n);
}

```
for (i = 3; i*i <= n; i += 2)
{   while (n%i == 0)
    { n /= i; cout << i << " ";
    }
}
if (n > 1)
{ cout << n
}
```

$i*i > n$

n = P_1 or P_2 or P_3

$9 < 11$

i = 4 4*4 = 16 > 11

```
for (i = 3; i*i <= n; i += 2)
{
        while (n%i == 0)
        { n /= i;
        }
}
```

$n = 5^2\ 7^2\ 11^2$

16 > n

25 =

n = 49

n = 1

25 =

for (i = 3; i*i <= n; i += 2)

{

while (n % i == 0)

{ n /= i;

}

}

n = 25

n = 1

5*5 <= 25

n = 1

i*i > n

n = 1 or $(P_1)^1 (P_2)^1 (P_3)^1$

= 1 or n (Prime)

45 i = 3 45 = 5

i = 4 × 4 = 16 > 5

84 → 7 1

P¹

i = 4 4×4 = 16 > 7

100 = 2×2×5×5

3 4 5 ×5 = 25 == 25

6 ×6 = 36 > 1

T.C = O(√n)

N   1 – N   **No of Prime Nos?**

check_Prime()

$\sqrt{n} * \sqrt{n}$

```
for (i=3; i<=n; i+=2)
{
    if (check_Prime(i))
    { cout<<i
    }
}
```

$\sqrt{3} + \sqrt{5} + \sqrt{7} + \sqrt{11} - - \sqrt{n}$

$n\sqrt{n}$

Optimize?

Seive of Eratosthenes

1 – n

```
vector<bool> Prime(n,1)
for ( i=3; i<N; i+=2)
{
    if (isPrime[i])
    {
        for (j=2*i; j<n; j+=i)
        {
            isPrime[j]=false
        }
    }
}
```

i=2   i=2   n>i

Prime NO: 1, N   2 – N-1

1 – N   Prime   2 – N   T.C = ?

n/2, n/3, n/4

n=10

1 → 4, 6, 8, 10

n/2 = 5

Composite

4, 8, 12, 16, 20, 24

2

Prime NO, Composite No

6

6, 12, 18, 24, 30 - -

2 3

n/2

n

100

2 4 6 8 10

50

100 n=3

33

3 6 9 12

33*3 = 99

$\frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \frac{n}{6} - - -$

$n\left[\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + - -\right]$

log n

$\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7} + \frac{n}{11} + - -$

$n\left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + - -\right]$

n(log(log(n))

O(log(log(n)))

i=i+1

HP[ ]

40

7*7

40

i=i

Highest Prime factor

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 31 32 33 34

21 ~~22~~ ~~23~~ 24 ~~25~~ 26 ~~27~~ ~~28~~ 29 ~~30~~ 31 ~~32~~ ~~33~~ ~~34~~

~~35~~ 36 37 ~~38~~ ~~39~~ ~~40~~

$2*3=6$ $i*i$ $5*5$ $7*7=49$

$2*i$

$2*i$ ⇒ NU ⇒ even Multiple of 2

$\frac{n}{2} = \frac{2*i}{2}$

$3*i$ $5*i$ odd

unmarked ⇒ i^2 i $i*i>n$ $i*i>n$

$2*i$ $3*i$ $4*i$ $5*i$ $i*i$ $(i-1)*i$

```
for (i=3; i*i<=n; i+=2)
  if (isPrime[i])
    for (j = i*i; j<=n; j+=i)
      {
      }
```

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

~~Prime~~ Prime Nos upto n (N)

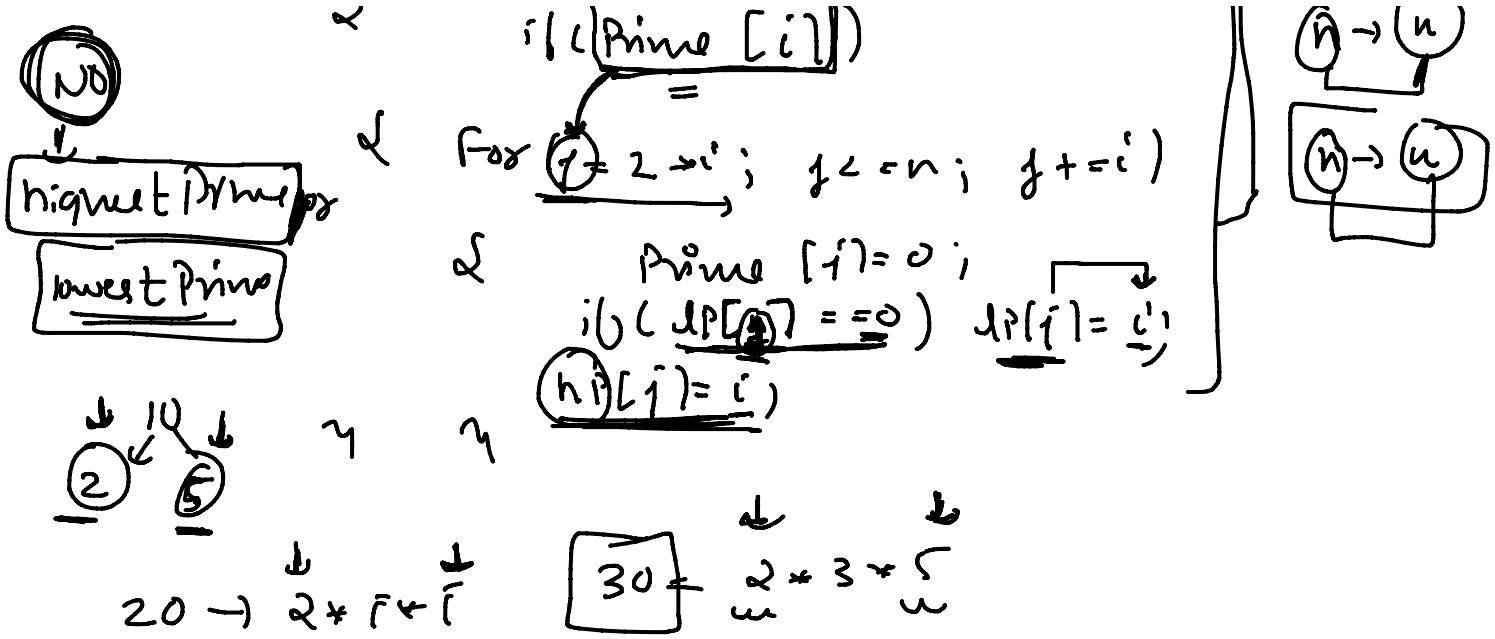
Sieve of Eratosthenes

Queries (Q) →

```
vector<int> hP(n,0); vector<int> dP(n,0)
for (i=3; i<=n; i+=2)
  if (Prime[i])
```

(N0)

n → n


NO
highestPrime
lowestPrime
if(!Prime[i])
{
For (j=2*i; j<=n; j+=i)
{
Prime[j]=0;
if(dp[j]==0) dp[j]=i;
hi[j]=i;
}
}
n → u
n → u
10
2
5
20 → 2*2*5
30 → 2*3*5