

PROGRAM 06: Optimising Sieve of Eratosthenes

Ex: 1 $N = 21$

STEP:01

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	21
F	F	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	...	T

$i=2$ $N+1$

STEP:02

Outer loop optimization

i prime

2 } True

3 } True

5 } True

7 } True

$j = i * i$ — $(j \leq N)$ — Inner loop optimization

Non-Prime

4 6 8 10 12 14 16 18 20

9 12 15 18 21

25 \leq 21 $\times \rightarrow (i * i \leq N)$

49 \leq 21 $\times \rightarrow (i * i \leq N)$

False

True \Rightarrow 2 3 5 7 11 13 17 19 \Rightarrow total prime = 8
 False \Rightarrow 2 4 6 8 10 12 14 16 18 20 9 15 21 \Rightarrow non-prime
 $\Rightarrow 21 - 8 = 13$

① inner loop $(j = i * i \text{ ————— } j \leq N)$
 $j = i^2 \text{ ————— } N$
 ② outer loop $(i = 2 \text{ ————— } i * i \leq N)$
 $i = 2 \text{ ————— } \sqrt{N}$

```
// PROGRAM 06: Optimising Sieve of Eratosthenes
```

```
#include<iostream>
```

```
#include<vector>
```

```
#include <cmath>
```

```
using namespace std;
```

```
// ☒ Optimising Sieve of Eratosthenes --> T.C. =  $O(N * (\log(\log N)))$  and S.C. =  $O(N)$ 
```

```
vector<bool> Sieve(int N){
```

```
    // Create the sieve array of N size telling isPrime or not
```

```
    vector<bool> sieve(N+1,true);
```

```
    sieve[0]=sieve[1]=false;
```

```
    // Optimization 02: replace  $i \leq N$  with  $i*i \leq N$  in outer loop
```

```
    for(int i=2; i*i<=N ; i++){
```

```
        if(sieve[i]){
```

```
            // Optimization 01: replace  $j = i * 2$  with  $j = i * i$  in inner loop
```

```
            // first unmarked number would be  $i*i$ , as others have been marked by 2 to  $(i-1)$ 
```

```
            for(int j=i*i; j<=N; j+=i){
```

```
                sieve[j]=false;
```

```
            }
```

```
        }
```

```
    }
```

```
    return sieve;
```

```
}
```

```
int main(){
```

```
    int N = 21;
```

```
    vector<bool> sieve = Sieve(N);
```

```
    for(int i=0;i<=N;i++){
```

```
        if(sieve[i]){
```

```
            cout<<i<<" ";
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Outer loop

Inner loop

$T.C \Rightarrow O(N * (\log(\log N)))$