Complexity Analysis

In this lesson, we'll see the run-time analysis for Sieve.

The outer loop

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

Runs for $\sqrt{N}$ values but the number of iterations of the inner loop is dependent on i.

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

Also, note that the inner loop only runs for values when i is prime.

i	**Inner loop iterations**
2	$\frac{N}{2}$
3	$\frac{N}{3}$
4	0
5	$\frac{N}{5}$
6	0
7	$\frac{N}{7}$
8	0

Total number of operations:

$$\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \dots$$

$$N[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \dots]$$

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

The growth of the sum of the reciprocals of primes is $O(log(logN))$. Proof is beyond the scope of this lesson but folks who would like to know more can go here.

Growth of sum of reciprocals of primes is $O(log(logN))$. Proof is beyond the scope but interested folks can go here

This gives us the total time complexity of Sieve of Eratosthenes - $O(N * log(logN))$.

In the next lesson, we'll see a variation of Sieve for a higher range of integers.