Sieve of Eratosthenes

In this lesson, we'll see an efficient algorithm to generate prime numbers.

We'll cover the following

Generating primes

Given an integer N, you are asked to print all the prime numbers between 1 and N.

Using what we have discussed so far, one way to do this would be to iterate over all the numbers from 1 to N and check if it's prime or not.

Time Complexity - $O(N * \sqrt{N})$.

It should be okay for N up to 10^4 or even 10^5 but not more.

Sieve of Eratosthenes

The Sieve of Eratosthenes is a simple algorithm to generate all primes to generate all primes from 1 to N in $O(N * log(logN))$.

Steps:

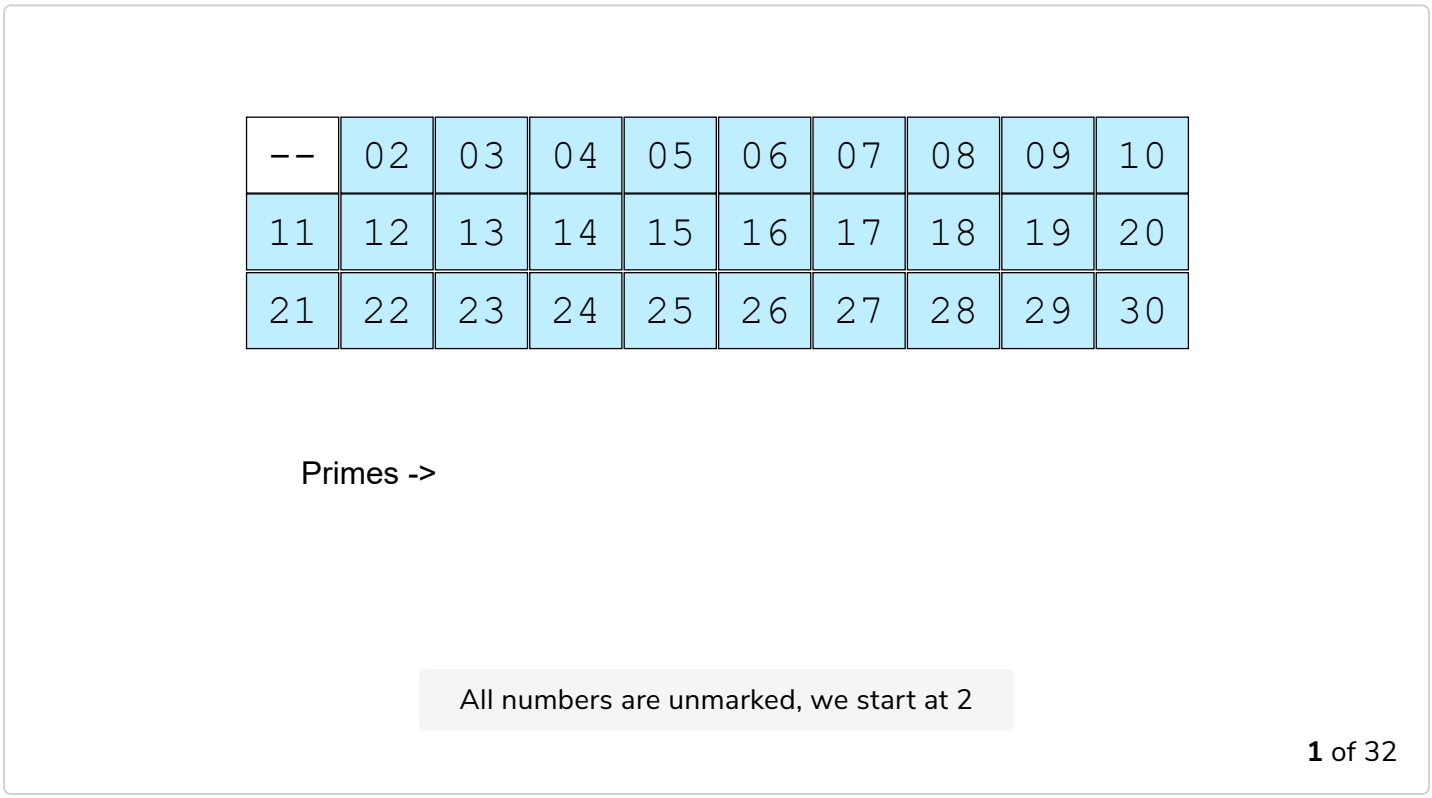
1. Create a list of all numbers from 2 to N. Initially, all numbers are unmarked.
2. Starting from $p = 2$, we will mark all multiples of 2 less than or equal to N. These numbers are definitely not prime since 2 divides them.
3. Move to the next unmarked number, i.e., $p = 3$. Mark all its multiples.
4. Stop if $p > \sqrt{N}$

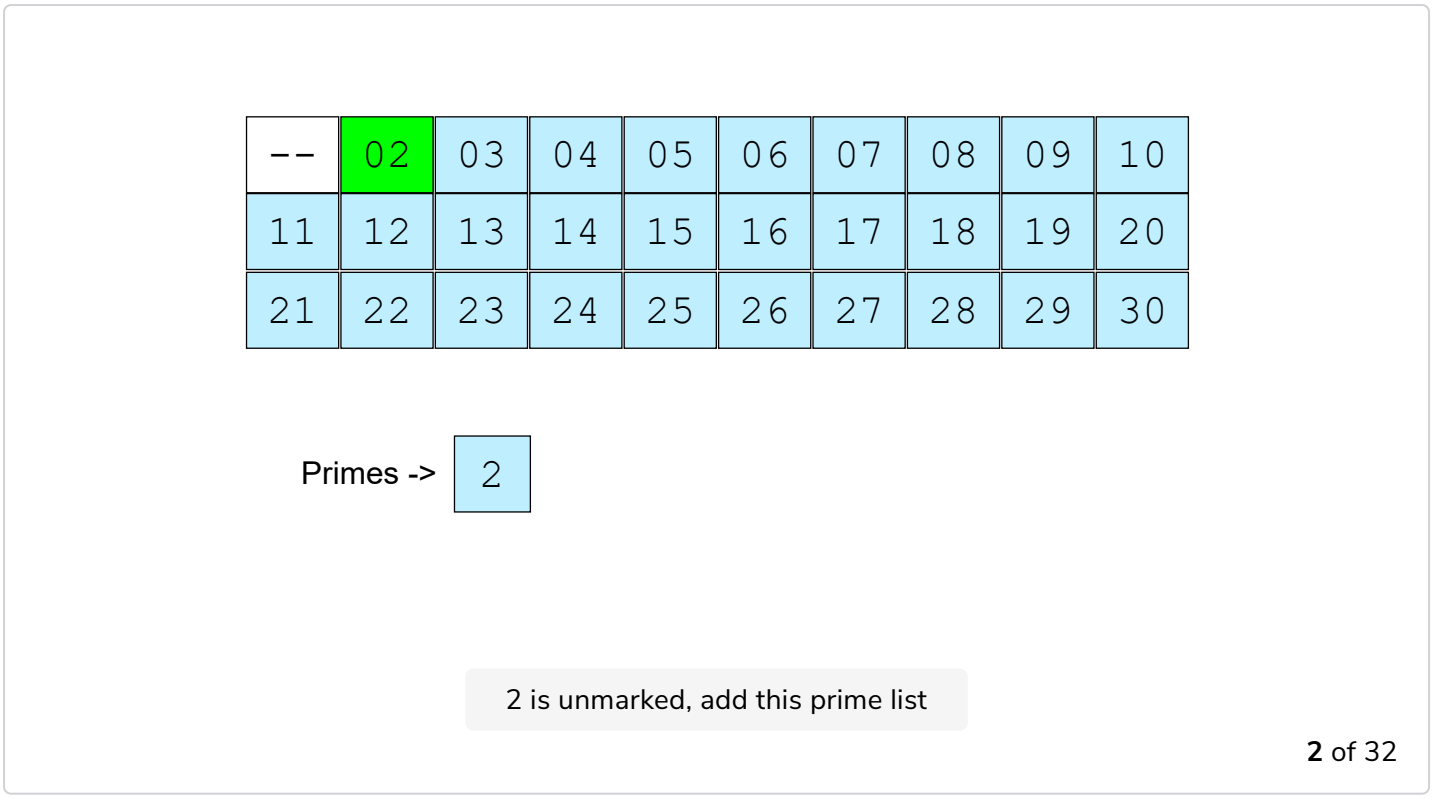
Each unmarked number that we visit is a prime because all non-primes will be

marked by one of its factors before we reach this number.

We can use a Boolean array of size N to distinguish between marked and unmarked numbers.

Let's understand the process better using the illustration below for $N = 30$. We will stop after we reach $ceil(\sqrt{30})$ i.e 6





--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

mark all multiple of 2 as non-prime

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3

Find the next unmarked number i.e 3. Add to Primes list

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3

Mark all multiples of 3 as non-prime

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3

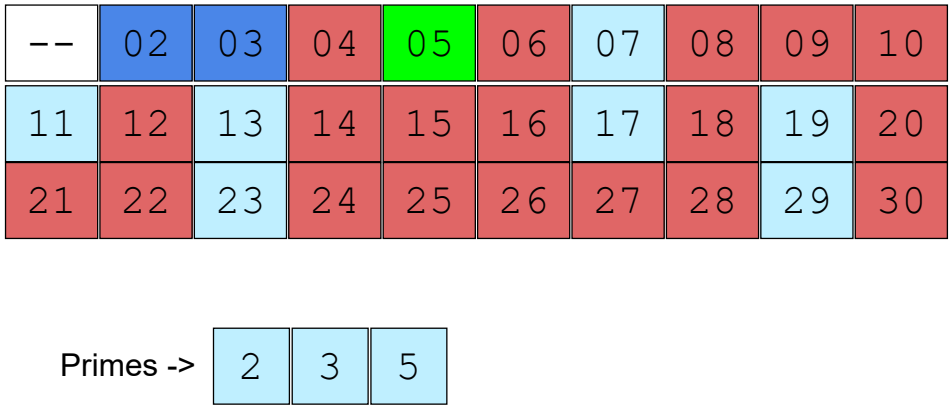
--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3

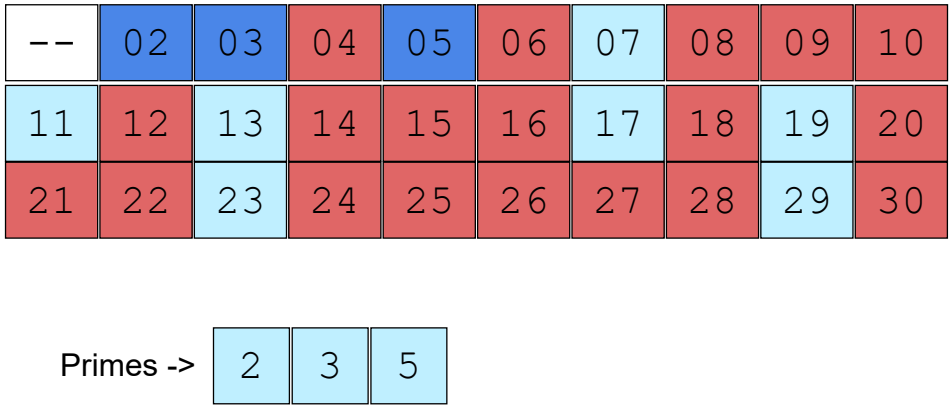
--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3 5

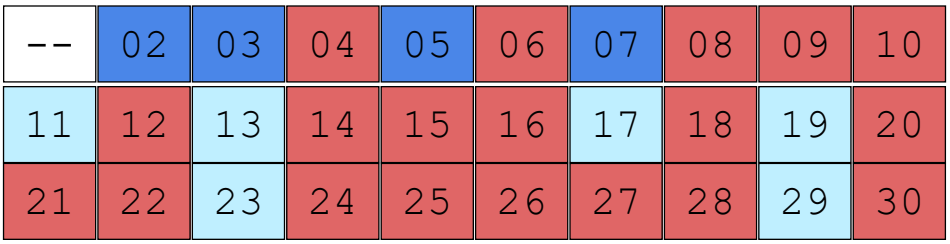
Find next unmarked number i.e 5. Add to Primes list



Mark all multiple of 5 non-prime

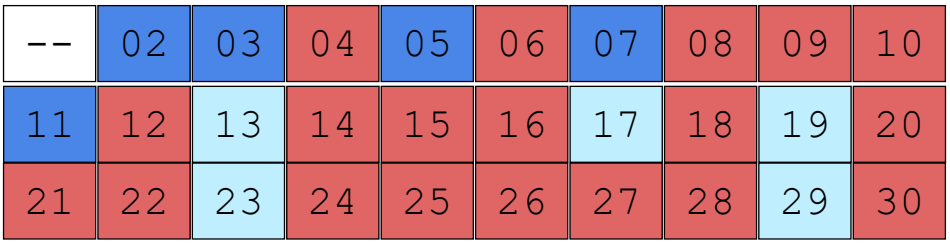


We don't iterate after 6. All remaining unmarked numbers are also prime



Primes -> 2 3 5 7

We don't iterate after 6. All remaining unmarked numbers are also prime



Primes -> 2 3 5 7 11

We don't iterate after 6. All remaining unmarked numbers are also prime

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes ->	2	3	5	7	11	13

We don't iterate after 6. All remaining unmarked numbers are also prime

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes ->	2	3	5	7	11	13	17

We don't iterate after 6. All remaining unmarked numbers are also prime

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

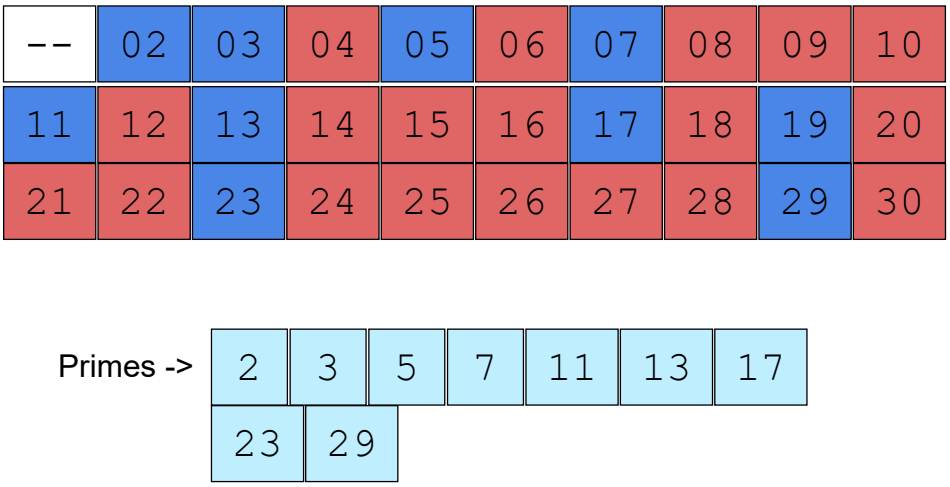
Primes -> 2 3 5 7 11 13 17 19

We don't iterate after 6. All remaining unmarked numbers are also prime

--	02	03	04	05	06	07	08	09	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Primes -> 2 3 5 7 11 13 17 23

We don't iterate after 6. All remaining unmarked numbers are also prime



We don't iterate after 6. All remaining unmarked numbers are also prime

We stop after $ceil(\sqrt{N})$ because for any non-prime $x <= N$ has a factor f such that $f <= \sqrt{N}$ as previously discussed.

So, for any number between 1 and 30 that is not a prime has a factor $f <= 6$.

In the lesson, we'll see how to implement this algorithm.