Implementation

In this lesson, we'll see the implementation of Sieve of Eratosthenes.

We'll cover the following

Implementation

The Boolean array $iis\_prime[]$ of Boolean is initialized to $true$ and denotes that all numbers are assumed prime initially and marks them $false$ as we proceed.

We start with $i = 2$ and only go up to $ceil(\sqrt{N})$ as explained in previous chapter.

> Instead of comparing as $i <= \sqrt(N)$, we are comparing as $i * i <= N$. It's essentially the same comparison and we avoid dealing with the floating-point variable.

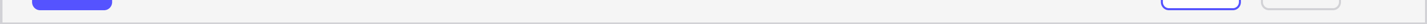
```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
  int N = 30;
  vector<bool> is_prime(N + 1, true); // size is N+1 so we can access is_prime[N]

  for (int i = 2; i * i <= N; i++) {
    if (is_prime[i]) {
      for (int j = i + i; j <= N; j += i)
        is_prime[j] = false;
    }
  }

  for (int i = 2; i <= N; i++)
    if (is_prime[i])
      cout << i << " ";

  return 0;
}
```



In the next lesson, we’ll calculate the run-time complexity of Sieve.