Solved Problem - Segmented Sieve

In this lesson, we'll discuss how to implement the segmented Sieve: a very popular variation of sieve.

We'll cover the following

Problem statement

Given two integers, N and M, print all primes between N and M (*both inclusive*).

Input format

The only line of input contains two integers N and M $(1 \leq N \leq M \leq 10^9)(M - N \leq 10^6)$.

Sample

Input

```
100000000 100000100
```

Output

```
100000007 100000037 100000039 100000049 100000073 100000081
```

Sieve

Obviously, Sieve of Eratosthenes comes to mind. Now, two things don't work right here:

1. $M <= 10^9$, we don't have enough memory to declare an array of a billion integers (4 GB memory).
2. Even if we could declare an array that big, the time complexity would be $O(M * log(logM))$, which is obviously very slow.

Observation: $M - N <= 10^6$

Segmented sieve

Since the range in which we want the generate the primes is at max 10^6, we can run a modified version of sieve on a Boolean array $A[]$ of size $M - N + 1$ such that.

- $A[0]$ - denotes whether N is prime or not.
- $A[1]$ - denotes whether $N + 1$ is prime or not.
- ...
- $A[M - N]$ - denotes whether M is prime or not.

Marking multiples not-prime

Comparing to sieve where we iterate up to $\sqrt{N}$ and if the current number is prime, we mark its multiples not-prime.

Here, we will iterate up to $\sqrt{M}$, but if the number doesn't belong in the range $[N, M]$, we don't know if it's a prime or not. So, in this case, we will mark multiples not-prime for **all** the numbers in $[2, \sqrt(M)]$.

We only need to mark the multiples if the multiple is between N and M. To do this, we start with the first multiple of this number that is greater than or equal to N.

First multiplex of x just greater than or equal to N can be calculated as follow:

$m = (\lfloor \frac{N-1}{x} \rfloor + 1) * x$

main.cpp

input.txt

All code files are copied to end of the page...

Time complexity

What's the order and count of operations? For each i in $[2, \sqrt{M}]$, we iterate over its multiple in $[N, M]$.

- $i = 2$, $\frac{M-N}{2}$ multiples
- $i = 3$, $\frac{M-N}{3}$ multiples
- $i = 4$, $\frac{M-N}{4}$ multiples
- ...
- $i = \sqrt{M}$, $\frac{M-N}{\sqrt{M}}$ multiples

Number of operations:

$$(M - N) \times [\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ... + \frac{1}{\sqrt{M}}]$$

As discussed in the complexity analysis chapter, the second term is a harmonic series with an upper bound of $logM$.

Time complexity - $O((M - N) * logM)$

In the next lesson, we'll start with string manipulation methods and problems.

Code Files Content !!!

⤋ ⤋

```
-----------------------------------------------------------------------
|  main.cpp [1]
-----------------------------------------------------------------------
```

```
#include

#include
#include
using namespace std;

int main() {
  ifstream cin("input.txt");

  int N, M;
  cin >> N >> M;
  vector is_prime(M-N+1, true);

  for (int i = 2; i * i <= M ; i++) {
    int start = (((N - 1) / i) + 1) * i;
    for (int j = start; j <= M ; j+= i) {
      if (j >= N && j <= M)
        is_prime[j - N] = false;
    }
  }

  for (int i = 0; i < is_prime.size(); i++) {
    if (is_prime[i])
      cout << i + N << " ";
  }

  return 0;
}


-------------------------------------------------------------------------
|  input.txt [1]
-------------------------------------------------------------------------


100000000 100000100




*********************************************************************************
```