March 13, 2019

Homework 1: Solution (Sieve of Eratosthenes)
50 Points

Consider the algorithm named Sieve of Eratosthenes:

Sieve
Input: an integer $n > 1$.
Let A be an array of Boolean values, indexed by integers 2 to n, initially all set to true.
for $i = 2, 3, 4, \ldots$, not exceeding $\sqrt{n}$:
    if $A[i]$ is true:
        for $j = i^2, i^2 + i, i^2 + 2i, i^2 + 3i, \ldots$, not exceeding n:
            $A[j] :=$ false.
Output: all i such that $A[i]$ is true.

Here is C++ code implementing the algorithm:

```
void Sieve(fsu::BitVector& bv)
// in:  bv is a bit vector
// out: bv[k] is true iff k is prime
{
  // find max and sqrt(max)
  size_t max = bv.Size();
  size_t sqrtmax = ceil(sqrt(max));

  // 1: initialize bv
  bv.Set();     // set all bits
  bv.Unset(0);  // 0 is not prime
  bv.Unset(1);  // 1 is not prime
  // 2: unset all even bits > 2
  for (size_t i = 4; i < max; i += 2)
  {
    bv.Unset(i);
  }
  // 3: clear bits at multiples of all odd primes < sqrt(max)
  for (size_t i = 3; i < sqrtmax; i += 2)   // we can skip over the even numbers, already unset
  {
    if (bv[i]) // i is prime
    {
      for (size_t j = i*i; j < max; j+= i)  // clear all multiples of i
      {
        bv.Unset(j);
      }
    }
  }
}
```

Along with Euclid's algorithm to find the greatest common divisor of two positive integers, the Sieve of Eratosthenes is one of the oldest computational algorithms, dating back at least to 300 BC [1], [2]. The Sieve is still an important tool today, so much so that it has been improved and optimized in many ways.

One modern version may help discover new non-Mersenne primes [3]. (The current record prime numbers are all Mersenne primes, that is, of the from $2^p - 1$ for some much smaller prime p. These discoveries used techniques that work only for the rare Mersenne primes. See [4].)

Your task is to prove:

Proposition. The Sieve of Eratosthenes algorithm stated above has runtime complexity $\leq \mathcal{O}(n \log \log n)$.

This may seem a daunting assignment at first. However, please read all about the Sieve in Wikipedia and any other open sources on the web. You may assume some facts about prime numbers and logarithms as well, including:

Prime Number Theorem. Let $\pi(x)$ denote the number of primes less than or equal to x. Then:

$$\pi(x) \sim \frac{x}{\log x}$$

where $\log x$ is the natural logarithm of x. [5]

and the result of Mertens (1874) that *pre-dates* the prime number theorem by 22 years:

Mertens' Lemma. Take $\sum_{p \leq n}$ to mean the sum over all prime numbers $p \leq n$. Then

$$\sum_{p \leq n} \frac{1}{p} \sim \log \log n + M$$

where $M \approx 0.261497...$ is the so-called *Meissel-Mertens constant*. [6]

To prove the Proposition, first eliminate the optimizations to obtain a simplified version of Sieve (see below). Show in passing that the total loop count of Sieve(n) is $\leq$ that of Simpler Sieve(n) and then count the steps in the loops. In particular, show that the number of steps in the inner loop executing at the prime p is $\leq \frac{n}{p}$. Then find the total count of both loops, do a little algebraic re-arranging, and apply Mertens' Lemma. (You supply full details ... this is just a hint.)

Simpler Sieve (Eratosthenes' original verson)

Input: an integer $n > 1$.
Let A be an array of Boolean values, indexed by integers 2 to n, initially all set to true.
for $i = 2, 3, 4, \ldots$, not exceeding n:
    if $A[i]$ is true:
        for $j = i + i, i + 2i, i + 3i, \ldots$, not exceeding n:
            $A[j]$:= false.
Output: all i such that $A[i]$ is true.

Cite your sources!

References

[1] According to Wikipedia, the earliest known reference to the sieve is in Nicomachus of Gerasa's Introduction to Arithmetic (translated from ancient Greek in [2]) which describes it and attributes it to Eratosthenes of Cyrene, a Greek mathematician.

[2] Hoche, Richard, ed. (1866), Nicomachi Geraseni Pythagorei Introductionis arithmeticae libri II, Leipzig: B.G. Teubner, p. 31

[3] See Science Alert: https://www.sciencealert.com/an-ancient-greek-algorithm-could-be-the-key-to-finding-new-prime-numbers

[4] See GIMPS: https://en.wikipedia.org/wiki/Great_Internet_Mersenne_Prime_Search

[5] See the Wikipedia entry: https://en.wikipedia.org/wiki/Prime_number_theorem

[6] F. Mertens. **J. reine angew. Math. 78** (1874), 46 *Ein Beitrag zur analytischen Zahlentheorie*

Solution

Lemma. The loop

for $j = p + p, p + 2p, p + 3p, \ldots$, not exceeding n

executes its body at most $\frac{n}{p}$ times.

Proof of Lemma. The loop begins at index $j = 2p$ and terminates at the last index $j = kp$ such that $kp \leq n$. Manipulating the inequality we have $k \leq \frac{n}{p}$ as an upper bound on k. Therefore the loop header is equivalent to:

for $k = 2, 3, 4, \ldots$, not exceeding $\frac{n}{p}$

by setting $j = kp$ in the loop body. Plainly this loop executes its body $\leq \frac{n}{p}$ times. □

Proof of Proposition. Inspecting the loop structures, we see that Sieve(n) has runtime bounded above by the runtime of SimpleSieve(n) because the loops just run longer in the simple version. For the remainder of this proof we concentrate on the simple version and use the notation $S(n)$ to mean the number of atomic steps executed by simplified Sieve evaluated at n.

Applying the Lemma we see that the inner loop of the algorithm executes $\leq \frac{n}{i}$ times whenever i is prime and 0 times whenever i is not prime. The outer loop executes n times. Therefore the runtime is bounded above by the sum:

$$S(n) \leq \mathcal{O}(\sum_{p \leq n} \frac{n}{p})$$

where the sum is over all primes $\leq n$. Pulling the factor n out of the terms and applying Mertens' Lemma we have:

$$S(n) \leq \mathcal{O}(\sum_{p \leq n} \frac{n}{p}) = \mathcal{O}(n \sum_{p \leq n} \frac{1}{p}) = \mathcal{O}(n(\log \log n + M)) = \mathcal{O}(n \log \log n)$$

and the proof of this rather deep result is complete. □