Prime Numbers

2,3,5,7,…

Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Fall 2023

Objectives

- Implement the Sieve of Eratosthenes
- Factor 128 bit numbers
- Enumerate some applications of prime numbers

Method 1 — Trial Division

You need to see if a number is prime / factorize a number. How can you do that?

- Trial division…

```
pIsPrime = true;
for(i=2; i<p; ++i)
    if (p % i == 0) {
        pIsPrime = false;
        break;
    }
```

Method 2 — A Slight Improvement

- Improvement: only check the odd numbers

```
pIsPrime = true;
if (p % 2 == 0)
  pIsPrime = false;
else
  for(i=3; i<p; i+=2)
      if (p % i == 0) {
        pIsPrime = false;
        break;
      }
```

Method 3 — Stop at $\sqrt{p}$

- We can stop at $\sqrt{p}$.
- If $q > \sqrt{p}$ and $q|p$, then there is a factor $k < \sqrt{p}$ such that $kq = p$.

```
#include <cmath> // or bits/stdc++.h

int sqrtP = std::sqrt(p)
pIsPrime = true;
if (p % 2 == 0)
  pIsPrime = false;
else
  for(i=3; i<sqrtP; i+=2)
      if (p % i == 0) {
        pIsPrime = false;
        break;
      }
```

The Sieve

```
// From Competitive Programming 3
#include <bitset>
ll _sieve_size; // 10^7 should be enough for most cases
bitset<10000010> bs;
vi primes;

void sieve(ll upperbound) {
  _sieve_size = upperbound + 1;
  bs.set();  // all bits set to 1
  bs[0] = bs[1] = 0;
  for (ll i = 2; i <= _sieve_size; i++)
     if (bs[i]) { // cross out multiples from i * i!
        for (ll j = i * i; j <= _sieve_size; j += i)
           bs[j] = 0;
        primes.push_back((int)i);
} }
```

Factoring

- Once in a while you will be asked to factor a `long long int`, which has 128 bits.
  - These numbers can be up to 10^{18}.
  - To 10^9 there are 50,847,534 primes.
  - To 10^{18} there are 24,739,954,287,740,860 primes.