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.

Euclid's Algorithms

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

Your Objectives:

- Be able to calculate the GCD of two numbers using Euclid's algorithm.
- Use the extended Euclid's algorithm to solve Linear Diophantine equations.

Calculating the GCD

- Let $a > b > 0$.
- $gcd(a, b) = gcd(b, mod(a, b))$
- Why?

Calculating the GCD

- Let $a > b > 0$.
- $gcd(a, b) = gcd(b, mod(a, b))$
- Why?
- Fact 1: if $g|a$ and $g|b$ then $g|(a + b)$ and $g|(a - b)$
- So, we could use $gcd(a, b) = gcd(a - b, b)$

Calculating the GCD

- Let $a > b > 0$.
- $gcd(a, b) = gcd(b, mod(a, b))$
- Why?
- Fact 1: if $g|a$ and $g|b$ then $g|(a + b)$ and $g|(a - b)$
- So, we could use $gcd(a, b) = gcd(a - b, b)$
- That would be slow, so how about $gcd(a, b) = gcd(b, a - nb)$, where $n > 0 and a - nb > 0$ and minimal.

Calculating the GCD

- Let $a > b > 0$.
- $gcd(a, b) = gcd(b, mod(a, b))$
- Why?
- Fact 1: if $g|a$ and $g|b$ then $g|(a + b)$ and $g|(a - b)$
- So, we could use $gcd(a, b) = gcd(a - b, b)$
- That would be slow, so how about $gcd(a, b) = gcd(b, a - nb)$, where $n > 0 and a - nb > 0$ and minimal.
- Easy! Just let $n = mod(a, b)$

An example

$$\begin{aligned} gcd(a,b) = gcd(b, mod(a,b)) &= gcd(90,25) \\ &= gcd(25,15) \\ &= gcd(15,10) \\ &= gcd(10,5) \\ &= gcd(5,0) \\ &= 5 \end{aligned}$$

Diophantine Equations

- A *Diophantine Equation* is a polynomial equation where we are only interested in integer solutions.
- Linear Diophantine equation: $ax + by = 1$,
- It doesn't have to be 1....
- Running example: Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?

How to do it

- We want: $ax + by = g$, where $g = gcd(a, b)$. We know a, b, and we calculate g. How can we get x and y?

How to do it

- We want: $ax + by = g$, where $g = gcd(a, b)$. We know a, b, and we calculate g. How can we get x and y?
- Suppose we had:

$$bx_1 + (a \mod b)y_1 = g$$

How to do it

- We want: $ax + by = g$, where $g = gcd(a, b)$. We know a, b, and we calculate g. How can we get x and y?
- Suppose we had:

$$bx_1 + (a \mod b)y_1 = g$$

- Then take $a \mod b = a - \lfloor \frac{a}{b} \rfloor * b$ This gives:

$$bx_1 + (a - \left\lfloor \frac{a}{b} \right\rfloor * b)y_1 = g$$

How to do it

- We want: $ax + by = g$, where $g = gcd(a, b)$. We know a, b, and we calculate g. How can we get x and y?
- Suppose we had:

$$bx_1 + (a \mod b)y_1 = g$$

- Then take $a \mod b = a - \lfloor \frac{a}{b} \rfloor * b$ This gives:

$$bx_1 + (a - \left\lfloor \frac{a}{b} \right\rfloor * b)y_1 = g$$

- Rearrange a bit..

$$bx_1 + ay_1 - \left\lfloor \frac{a}{b} \right\rfloor by_1 = g \quad \Rightarrow \quad ay_1 + b(x_1 - \left\lfloor \frac{a}{b} \right\rfloor y_1) = g$$

How to do it

- We want: $ax + by = g$, where $g = gcd(a, b)$. We know a, b, and we calculate g. How can we get x and y?
- Suppose we had:

$$bx_1 + (a \mod b)y_1 = g$$

- Then take $a \mod b = a - \lfloor \frac{a}{b} \rfloor * b$ This gives:

$$bx_1 + (a - \left\lfloor \frac{a}{b} \right\rfloor * b)y_1 = g$$

- Rearrange a bit..

$$bx_1 + ay_1 - \left\lfloor \frac{a}{b} \right\rfloor by_1 = g \quad \Rightarrow \quad ay_1 + b(x_1 - \left\lfloor \frac{a}{b} \right\rfloor y_1) = g$$

- This in turn gives us:

$$\begin{aligned} x &= y_1 \\ y &= x_1 - \lfloor \tfrac{a}{b} \rfloor y_1 \end{aligned}$$

The Code

$$x = y_1$$
$$y = x_1 - \left\lfloor \frac{a}{b} \right\rfloor y_1$$

```
// Stolen from Competitive Programming 3
// store x, y, and d as global variables
void extendedEuclid(int a, int b) {
   if (b == 0) { x = 1; y = 0; d = a; return; }
   extendedEuclid(b, a % b);
   // similar as the original gcd
   int x1 = y;
   int y1 = x - (a / b) * y;
   x = x1;
   y = y1;
}
```

An Example

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?

a	**b**	**x**	**y**	$a \times x + b \times y = 3$
72	33			
33	6			
6	3			
3	0			

An Example

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?

a	**b**	**x**	**y**	$a \times x + b \times y = 3$
72	33			
33	6			
6	3			
3	0	1	0	$3 \times 1 + 0 \times 0 = 3$

An Example

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?

a	**b**	**x**	**y**	$a \times x + b \times y = 3$
72	33			
33	6			
6	3	0	1	$6 \times 0 + 3 \times 1 = 3$
3	0	1	0	$3 \times 1 + 0 \times 0 = 3$

An Example

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?

a	**b**	**x**	**y**	$a \times x + b \times y = 3$
72	33			
33	6	1	-5	$33 \times 1 + 6 \times -5 = 3$
6	3	0	1	$6 \times 0 + 3 \times 1 = 3$
3	0	1	0	$3 \times 1 + 0 \times 0 = 3$

An Example

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?

a	**b**	**x**	**y**	$a \times x + b \times y = 3$
72	33	-5	11	$72 \times -5 + 33 \times 11 = 3$
33	6	1	-5	$33 \times 1 + 6 \times -5 = 3$
6	3	0	1	$6 \times 0 + 3 \times 1 = 3$
3	0	1	0	$3 \times 1 + 0 \times 0 = 3$

An example, ctd.

- Suppose you go to the store. You buy *x* apples at 72 cents each and *y* oranges at 33 cents each. You spend \$5.85. How many of each did you buy?
- Running the algorithm, we get...

$$72 \times -5 + 33 \times 11 = 3$$

- We multiple both sides by 195 (since $585 = 3 \times 195$) This gives us...

$$72 \times -975 + 33 \times 2145 = 585$$

An example, ctd.

- Suppose you go to the store. You buy *x* apples at 72 cents each and *y* oranges at 33 cents each. You spend \$5.85. How many of each did you buy?
- Running the algorithm, we get...

$$72 \times -5 + 33 \times 11 = 3$$

- We multiple both sides by 195 (since $585 = 3 \times 195$) This gives us...

$$72 \times -975 + 33 \times 2145 = 585$$

- We can add $(\frac{33}{3} = 11)n$ to the 72 term and subtract $(\frac{72}{3} = 24)n$ from the second term and still have a valid equation.

An example, ctd.

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?
- Running the algorithm, we get...

$$72 \times -5 + 33 \times 11 = 3$$

- We multiple both sides by 195 (since $585 = 3 \times 195$) This gives us...

$$72 \times -975 + 33 \times 2145 = 585$$

- We can add $(\frac{33}{3} = 11)n$ to the 72 term and subtract $(\frac{72}{3} = 24)n$ from the second term and still have a valid equation.
- Solve $-975 + 11n > 0$, this reduces to $n > 88.6$. So take $n = 89$.

An example, ctd.

- Suppose you go to the store. You buy x apples at 72 cents each and y oranges at 33 cents each. You spend \$5.85. How many of each did you buy?
- Running the algorithm, we get...

$$72 \times -5 + 33 \times 11 = 3$$

- We multiple both sides by 195 (since $585 = 3 \times 195$) This gives us...

$$72 \times -975 + 33 \times 2145 = 585$$

- We can add $(\frac{33}{3} = 11)n$ to the 72 term and subtract $(\frac{72}{3} = 24)n$ from the second term and still have a valid equation.
- Solve $-975 + 11n > 0$, this reduces to $n > 88.6$. So take $n = 89$.
- This gives us the final equation

$$72 \times 4 + 33 \times 9 = 585$$