CSE 311 Foundations of Computing I

Lecture 12

GCD and Euclid's Algorithm

Autumn 2011

Announcements

- Reading assignments
  - Today :
    - 7th Edition: 4.3 (the rest of the chapter is interesting!)
    - 6th Edition: 3.5, 3.6
    - 5th Edition: 2.5, 2.6 up to p. 191
  - Wednesday
    - 7th Edition: 5.1, 5.2
    - 6th Edition: 4.1, 4.2
    - 5th Edition: 3.3, 3.4
- Updated slides from lectures 10 and 11 have been posted
- Paul out of town Monday, Richard out of town Friday

Highlights from last lecture

- Applications of modular arithmetic
- Modular exponentiation
- Primality

Fast exponentiation

```
namespace _311ConsoleApp {
    class Program {
        static void Main(string[] args) {
            FastExp(2, 16, 10000);
            System.Console.ReadLine();
        }

        static int FastExp(int x, int n, int modulus) {
            long v = (long)x;
            int exp = 1;
            for (int i = 1; i <= n; i++) {
                v = (v * v) % modulus;
                exp = exp + exp;
                System.Console.WriteLine("i: " + i
                    + "  exponent: " + exp + "  v: " + v);
            }
            return (int)v;
        }
    }
}
```

```
file:///C:/Users/Richard/D...
i: 1  exponent: 2  v: 4
i: 2  exponent: 4  v: 16
i: 3  exponent: 8  v: 256
i: 4  exponent: 16  v: 5536
i: 5  exponent: 32  v: 7296
i: 6  exponent: 64  v: 1616
i: 7  exponent: 128  v: 1456
i: 8  exponent: 256  v: 9936
i: 9  exponent: 512  v: 4096
i: 10  exponent: 1024  v: 7216
i: 11  exponent: 2048  v: 656
i: 12  exponent: 4096  v: 336
i: 13  exponent: 8192  v: 2896
i: 14  exponent: 16384  v: 6816
i: 15  exponent: 32768  v: 7856
i: 16  exponent: 65536  v: 6736
```


Primality

An integer *p* greater than 1 is called *prime* if the only positive factors of *p* are 1 and *p*.

A positive integer that is greater than 1 and is not prime is called composite.

Every positive integer greater than 1 has a unique prime factorization

Euclid's theorem

- There are an infinite number of primes.
- Proof by contradiction:

  Suppose there are a finite number of primes: $p_1, p_2, \ldots, p_n$

Distribution of Primes

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173
179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263
269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359

- If you pick a random number n in the range [x, 2x], what is the chance that n is prime?

Famous Algorithmic Problems

- Primality Testing:
  - Given an integer n, determine if n is prime
- Factoring
  - Given an integer n, determine the prime factorization of n

Factoring

- Factor the following 232 digit number [RSA768]:

123018668453011775513049495838496272077285356959
533479219732245215172640050726365751874520219978
646938995647494277406384592519255732630345373154
826850791702612214291346167042921431160222124047
92747377940806653514195974595856902143413

123018668453011775513049495838496272077285356959
533479219732245215172640050726365751874520219978
646938995647494277406384592519255732630345373154
826850791702612214291346167042921431160222124047
92747377940806653514195974595856902143413

334780716989568987860441698482126908177047949837
137685689124313889828837938780022876147116525317
43087737814467999489

367460436667995904282446337996279526322791581643
430876426760322838157396665112792333734171433968
10270092798736308917

Greatest Common Divisor

- GCD(a, b): Largest integer d such that d|a and d|b
  - GCD(100, 125) =
  - GCD(17, 49) =
  - GCD(11, 66) =
  - GCD(180, 252) =

GCD, LCM and Factoring

$a = 2^3 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 = 46{,}200$

$b = 2 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 13 = 204{,}750$

$GCD(a, b) = 2^{\min(3,1)} \cdot 3^{\min(1,2)} \cdot 5^{\min(2,3)} \cdot 7^{\min(1,1)} \cdot 11^{\min(1,0)} \cdot 13^{\min(0,1)}$

$LCM(a, b) = 2^{\max(3,1)} \cdot 3^{\max(1,2)} \cdot 5^{\max(2,3)} \cdot 7^{\max(1,1)} \cdot 11^{\max(1,0)} \cdot 13^{\max(0,1)}$

Theorem

Let *a* and *b* be positive integers. Then

$a \cdot b = gcd(a, b) \cdot lcm(a, b)$

Euclid's Algorithm

- GCD(x, y) = GCD(y, x mod y)

Example: GCD(660, 126)

```
int GCD(int a, int b){   /* a >= b,   b > 0 */
      int tmp;
      int x = a;
      int y = b;
      while (y > 0){
            tmp = x % y;
            x = y;
            y = tmp;
      }
      return x;
}
```


Extended Euclid's Algorithm

- If GCD(x, y) = g, there exist integers s, t, such sx + ty = g;
- The values x, y in Euclid's algorithm are linear sums of a, b.
  - A little book keeping can be used to keep track of the constants

Bézout's Theorem

If *a* and *b* are positive integers, then there exist integers *s* and *t* such that

$gcd(a,b) = sa + tb$.

Simple cipher

- Caesar cipher, a → b, b → c, . . .
  - HELLOWORLD → IFMMPXPSME
- Shift cipher
  - f(x) = (x + k) mod 26
  - $f^{-1}(x)$ = (x – k) mod 26
- f(x) = (ax + b) mod 26
  - How good is the cipher f(x) = (2x + 1) mod 26

Multiplicative Cipher: f(x) = ax mod m

For a multiplicative cipher to be invertible:

f(x) = ax mod m : {0, m-1} → {0, m-1}

must be one to one and onto

Lemma: If there is an integer b such that ab mod m = 1, then the function f(x) = ax mod m is one to one and onto.

Multiplicative Inverse mod m

Suppose GCD(a, m) = 1

By Bézoit's Theorem, there exist integers s and t such that sa + tm = 1.

s is the multiplicative inverse of a:

$$1 = (sa + tm) \bmod m = sa \bmod m$$