CSE 321 Discrete Structures

February 1st, 2010

Lecture 12: Integer Division

Outline

- Quickly review set theory (see Lecture 6)
- The integers and division: read Rosen 3.4
- Andrew will discuss a homework problem

Announcement: Practice midterms will be posted later today

Number Theory (and applications to computing)

- Branch of Mathematics with direct relevance to computing
- Many significant applications
  - Cryptography
  - Hashing
  - Security
- Important tool set

Divisibility

Let a, b be two integers, and $a \neq 0$.
a ***divides*** b if there exists an integer c s.t. $a*c = b$

Notation: $a \mid b$

Divisbility

The Division "Algorithm". If a, d are integers and $d > 0$, then there exists unique q, r s.t.

(a) $0 \le r < d$ and

(b) $a = d*q + r$

a = dividend
d = divisor
q = quotient
r = reminder

q = a **div** d

r = a **mod** d

Primality

- An integer p is ***prime*** if its only divisors are 1 and p
- An integer that is greater than 1, and not prime is called ***composite***

Fundamental theorem of arithmetic:
Every positive integer greater than one has a unique prime factorization

Factorization

- If n is composite, it has a factor of size at most sqrt(n)

Euclid's theorem

There are an infinite number of primes.

Proof by contradiction:

- Suppose there are a finite number of primes: $p_1, p_2, \ldots p_n$
- Consider the number $p=1 + p_1 p_2 \ldots p_n$
  - Case 1: p is prime; contradiction
  - Case 2: p is not prime. Then it must be divisible by a prime number; but none of p_1, $p_2, \ldots p_n$; contradiction

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) =

Key properties:
- express GCD in terms of the prime factors
- GCD(a,b) = GCD(a-b, b) when a > b
- GCD(a,b) = GCD(r, b) when a mod b = r

Euclid's Algorithm

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

```
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;
}
```

% means mod in Java

How many steps ? In class…
(Ch.4.3, "Lame's Theorem")

Euclid's Algorithm

- A variant which uses only addition/ subtraction (no multiplication/division)

```
int GCD(int a, int b) {
    int x = a;   int y = b;

    while (x != y){
        if (x > y) x -= y;
        else y -= x;
    }
    return x;
}
```

Extended Euclid's Algorithm

- If GCD(x, y) = d, there exist integers s, t, such sx + ty = d;

```
int × int EGCD(int a, int b){ /* returns (s,t) */
    if (a == b) return (1,0);
    if (a>b) { (s,t) = EGCD(a-b, b);
               return (s, t-s); }
    else { (s,t) = EGCD(a, b-a);
               return (s-t, t); }
}
```

Prove correctness in class