MCIT 596
Euclid's Algorithm

GREATEST COMMON DIVISOR

- Given two positive integers a and b, find largest d that divides both.

- One solution: Factor a and b into prime factors… d is product of common factors.

- Example: $48 = 2^4 \times 3, \quad 90 = 2 \times 3^2 \times 5$: $GCD(48,90) = 2 \times 3 = 6$

(Note that 2^4 divides 48, but only 2^1 divides 90… So the highest power of 2 that divides both numbers is $\min(4,1) = 1$. Similar calculation needed for each prime factor.)

FACTORING

- To find $\gcd$ we need to factor numbers. How hard is that?
- Grade school algorithm again: To factor n, divide it by each i between 2 and $n - 1$. If you ever get 0 remainder, we have a factor.
- Can optimize further … try only numbers $< \sqrt{n}$ as divisors.
- Still: If n is a 1000-bit number, $\sqrt{n}$ is a 500-bit number, and there are roughly 2^{500} divisors you have to try. This takes time 2^{500} steps. More than the age of the universe in nanoseconds!
- Need an algorithm whose time does not grow exponentially in the number of bits. There are better algorithms for factoring, but they still don't cut it.

FACTS ABOUT GCDs

- Theorem: Suppose $a > b$. If $d = \gcd(a, b)$, then $d = \gcd(b, a - b)$.
- Proof:
- Any common factor of a and b is also a factor of $a - b$. Why?

  If x is factor of a and b, then $a = k_1 x$ and $b = k_2 x$ for integers k_1 and k_2.

  Then $a - b = (k_1 - k_2)\, x$.
- Similarly, any common factor of b and $a - b$ is also a factor of a.
- Thus the common factors of a and b are exactly the same as the common factors of b and $a - b$.
- Hence the $\gcd s$ of these two pairs of numbers are the same.

CONTINUING THIS OBSERVATION…

- Let $a = bq + r$ where q is the quotient, and $r < b$ is the remainder when dividing a by b.
- Then we can carry the idea of the previous slide further… repeatedly subtracting b, to see that $\gcd(a, b) = \gcd(b, r)$.
- When $r = 0$, we have found the $\gcd$: It is b.
- This is the idea of one of the oldest algorithms… Euclid's algorithm.

EUCLID'S ALGORITHM

- Input: Two positive numbers a and b, with $a > b$
  - While $((r = a \bmod b) > 0)$
    - $a \leftarrow b$
    - $b \leftarrow r$
  - Return b

- You can prove this algorithm is correct by using induction and the theorem we proved.

RUNNING TIME

- a and b are n-bit numbers. Let $T(n) =$ time taken by Euclid's algorithm.
- Consider 2 cases for first iteration of while loop:
  - If $b > \frac{a}{2}$, then, $r = a - b < \frac{a}{2}$
  - If $b < \frac{a}{2}$, then, $r < b < \frac{a}{2}$
  - Thus after one iteration r is at most half of a.
  - In second iteration b plays role of a and gets halved.
  - After 2 iterations both numbers are halved: i.e., both have at most $n - 1$ bits.
  - So $T(n) \leq 2 + T(n-1)$. Solution (by telescoping) is $T(n) = O(n)$.