

60-254, Lecture 1

60-254, Summer 2011

Dr. Asish Mukhopadhyay

Notion of an algorithm

- An *algorithm* is a tool for solving a well-specified computational problem.
- The problem describes in general terms the input/output relationship desired.
- An algorithm is one way of achieving this input/output relationship.

A simple problem

Problem:

Given two positive integers m and n , their greatest common divisor (gcd , for short) is the largest positive integer that divides both exactly.

Design an algorithm to determine this gcd efficiently.

Euclid's Algorithm

- **Algorithm** GreatestCommonDivisor

Input: Two positive integers m and n

Output: The **GCD** of m and n

Step1. Set $r \leftarrow m \bmod n$;

Step2. If $r \neq 0$, $m \leftarrow n$, $n \leftarrow r$, go to Step 1.

*Step3. Output n and **STOP**.*

Question ?

- How many times does Step 1 execute for a given pair of inputs m and n ($m > n$) ?

A little theorem

- **Theorem:** If $m > n$, then $m \bmod n < m/2$.
 - If $n \leq m/2$, $r = m \bmod n < n \leq m/2$
 - If $n > m/2$, $r = m - n < m/2$

Uncovering a pattern

- Consider the following sequence of remainders:

$$r_0 = m \bmod n < m/2$$

$$r_1 = n \bmod r_0 < n/2$$

$$r_2 = r_0 \bmod r_1 < r_0/2$$

$$r_3 = r_1 \bmod r_2 < r_1/2 < n/4$$

$$r_4 = r_2 \bmod r_3 < r_2/2 < r_0/4$$

•
•
•

- In 2 iterations the remainder becomes less than $n/2$, in 4 iterations less than $n/4$. Thus it will become 0 in at most $2 \log n$ iterations

Correctness of the GCD algorithm

- Is due to the loop invariant.

$$\text{gcd}(m,n)=\text{gcd}(n,r)$$

- Division of m by n can be written as:

$$m=q * n + r, \text{ where } 0 \leq r < n$$

- A divisor of m and n is a divisor of n and r and conversely.

Termination of the GCD algorithm

- The remainder r goes to 0 in a finite number of steps
- Why ?
 - sequence of remainders strictly decreases
 - is non-negative
 - must become zero in at most n steps.

Why Data Structures ?

- The subject of Data Structures is about different ways of organizing data
- The efficiency of an algorithm often depends on how the data is organized

An Example

- Given a sorted list of n numbers, find the middle element
- How should we store this list ?
 - Array
 - Linked list

Exercise

- Design an algorithm to determine if a list of n integers has repeated elements