Week 1 (1) – Introduction to Algorithms

CST370 – Design & Analysis of Algorithms

Dr. Byun

Computer Science

Lecture Objectives

- After completion of this lecture, you will be able to
  - explain the definition of algorithm.
  - determine the greatest common divisor (gcd) using three different approaches.
  - introduce a sieve of the Eratosthenes algorithm to generate prime numbers.

Chapter 1: Introduction

- **1.1 What is an Algorithm?**
- 1.2 Fundamentals of Algorithmic Problem Solving
- 1.3 Important Problem Types
- 1.4 Fundamental Data Structures

What is an algorithm?

- Watch the video first
  - https://youtu.be/qU_pPObygz8
  - You may want to turn on "**Closed Captions**".
- An ***algorithm*** is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.

The Notion of Algorithm

Problem – GCD Calculation

- The greatest common divisor of two nonnegative, not-both-zero integers *m* and *n*, denoted gcd(*m, n*), is defined as the largest integer that divides both *m* and *n* evenly, i.e., with a remainder of zero.
  - In the description, note that **input values, *m* and *n*, can't be zero at the same time**.

Example

- Calculate the gcd for the following four cases **with a pencil and paper**.

    // Do not move directly to the next slide.

    // First, solve this problem yourself.

    // Then, compare your answer with the solution

    // on the next slide.

1. gcd(0,0) = ?
2. gcd(6, 4) = ?
3. gcd(60,24) = ?
4. gcd(60,0) = ?

Example – Solution

- Watch this video https://youtu.be/HZ75wQ30uZ0

1. gcd(0,0) = Invalid input

    // By definition, both input values *n* and *m*

    // can't be zeroes in the gcd problem.

2. gcd(6, 4) = 2
3. gcd(60,24) = 12
4. gcd(60,0) = 60

    // Note that if one of two values *n* and *m* is 0,

    // another non-zero input value is the answer

    // to the gcd problem.

Euclid's Algorithm

- Watch this video first
  - https://youtu.be/XQSOqSegSVk
- Euclid's algorithm is based on repeated application of equality

$$\mathbf{gcd}(m, n) = \mathbf{gcd}(n, m \bmod n)$$

until the second number becomes 0, which makes the first number become the answer.

Euclid’s Algorithm – Example 1

- gcd(60, 24)

  = gcd(24, 60 mod 24)

  = gcd(24, 12)   // “60 mod 24” is 12.

  = gcd(12, 24 mod 12)

  = gcd(12, 0)   // “24 mod 12” is 0.

  = 12

Euclid’s Algorithm – Example 2

- gcd(4, 6)
  = gcd(6, 4 mod 6)
  = gcd(6, 4) // Note that “4 mod 6” is 4.
  = gcd(4, 6 mod 4)
  = gcd(4, 2)
  = gcd(2, 4 mod 2)
  = gcd(2,0)
  = 2

Exercise

- Calculate the **gcd(40, 56) using the Euclid's algorithm**.

  // Do not move directly to the next slide.

  // First, solve this problem yourself.

  // Then, compare your answer with the solution

  // on the next slide.

Exercise – Solution

- gcd(40, 56)
  = gcd(56, 40 mod 56)
  = gcd(56, 40)
  = gcd(40, 56 mod 40)
  = gcd(40, 16)
  = gcd(16, 40 mod 16)
  = gcd(16,8)
  = gcd(8, 16 mod 8)
  = gcd(8,0)
  = **8**

Pseudocode for Algorithm Description

- Pseudocode is **an informal high-level description** of an algorithm.
  - It uses the structural conventions of a normal programming language.
  - It typically omits details that are essential for machine understanding of the algorithm, such as variable declarations.
- **No standard for pseudocode** syntax exists.
  - In the class, we will try to follow the pseudocode convention in our textbook.

Example

ALGORITHM *Euclid*(m, n)

//Computes gcd(m, n) by Euclid's algorithm
//Input: Two nonnegative, not-both-zero integers m and n
//Output: Greatest common divisor of m and n
while $n \neq 0$ **do**
  $r \leftarrow m \bmod n$
  $m \leftarrow n$
  $n \leftarrow r$
return m

Pseudocode (1 of 2)

- The pseudocode in the textbook does not contain error handling used in the actual programming implementation.
  - Therefore, you should assume that the algorithms described in the textbook work only on the appropriate inputs.

Pseudocode (2 of 2)

- Read the following document to get the basic pseudocode notation of our textbook.
  - https://goo.gl/H88yR1

Pseudocode Example

```
Algorithm Average (A[0..n-1])
// Input: An array A with n numbers from
//          the index 0 to n-1
// Output: Average of the numbers in the array A
sum ← A[0]
for i ← 1 to n - 1 do
    sum ← sum + A[i]
avg ← sum / n
return avg
```

Two Other Methods to Calculate GCD

- Euclid’s algorithm works well to find gcd.
- However, there exist several other approaches to compute the gcd.

Consecutive integer checking algorithm for computing gcd (m, n)

Step 1 Assign the value of min{m,n} to t

Step 2 Divide m by t. If the remainder is 0, go to Step 3; otherwise, go to Step 4

Step 3 Divide n by t. If the remainder is 0, return t and stop;otherwise, go to Step 4

Step 4 Decrease t by 1 and go to Step 2

Middle-school procedure
for computing gcd (m, n)

Step 1 Find the prime factorization of m

Step 2 Find the prime factorization of n

Step 3 Find all the common prime factors

Step 4 Compute the product of all the common prime factors and return it as gcd(m,n)

Middle-school procedure – Example

- gcd (60, 24)
  1) 60 = 2 X 2 X 3 X 5
  2) 24 = 2 X 2 X 2 X 3
  3) gcd(60, 24) = 2 X 2 X 3 = 12.
     Note that **2, 2, and 3 are common** in 60 and 24.
- Is this an algorithm?
  – No because the prime factorization is not unambiguous.
  – Also, this approach is more complex and slower than Euclid's algorithm.

Some Important Points for Algorithms

- Input has to be specified carefully.
- The same algorithm can be represented in several different ways.
- Several algorithms for solving the same problem may exist.
- Algorithms for the same problem can be based on very different ideas and can solve the problem with dramatically different speeds.

Algorithmic Puzzle: Find a fake coin among eight coins

- There are eight identical-looking coins; one of these coins is counterfeit and is known to be **lighter than the genuine seven coins**.
- What is the **minimum number of weighings** needed to identify the fake coin with a two-pan balance scale without weights like below?

Important Note

- **Do not see the answer immediately** on the next page.
  - If you do, you will not get any knowledge from this puzzle.
- **Hint**: If your answer is three measurements, you have good computer science knowledge.
  - However, you can do better than that.

Solution (1 of 2)

- You can solve the puzzle by two measurements.
- First, select any six coins and put them on the scale (= 3 coins vs 3 coins).
  - If they weigh the same, the fake is among the other two coins.
  - Thus, weighing these two coins will identify the lighter fake.

Solution (2 of 2)

- If the first weighing of two groups (= 3 coins vs. 3 coins) does not yield a balance, the lighter fake is among the three lighter coins.
  - Take any two of them and put them on the scale (= 1 coin vs 1 coin).
  - If they weigh the same, it is the third coin in the lighter group that is fake; if they do not weigh the same, the lighter one is the fake.

Sieve of Eratosthenes

- This is an algorithm to identify prime numbers from 2 to a specific number ***n***.
- Example
  - Prime numbers to 10 are 2, 3, 5, and 7.
- To get the basic idea of the algorithm, watch this video
  - https://youtu.be/klcIklsWzrY

<<< Course Instruction >>>

- Read **from the page 3 to the page 7** of our textbook.
  - If you do not understand any part(s), study it again or ask it to the instructor.

<<< Course Instruction >>>

- This lesson is over.
  - If you have any questions, please contact your instructor.
- When you are done, study the next lecture (week_1_2.ppt) on Canvas.