

CSC 311: Design and Analysis of Algorithms¹

Tutorial 1

What is an algorithm?

An algorithm is a well-defined procedure composed of a sequence of unambiguous instructions for solving a well-defined computational problem. It takes a set of values as its input and produces another set of values as its output.

A computational problem is defined by describing the relationship between the input and the output.

A valid instance of a computational problem is a valid set of values representing the input of the algorithm. In order to be correct, an algorithm needs to be able to solve ALL valid instances of the problem (i.e., it halts and finds the right output).

Why do we study algorithms?

Simply because there are so many computational problems that need to be solved efficiently.

Can you name some?

Example:

Computational problem: Greatest Common Divisor (GCD).

Input: Two non-negative integers m and n , and at least one of m and n is greater than zero (i.e., $m > 0$ or $n > 0$).

Output: The GCD of m and n .

Algorithm 1: Finding the greatest common divisor.

Algorithm GCD(m,n)

while $n \neq 0$ **do**

$r := m \bmod n$;

$m := n$;

$n := r$;

end

return m ;

GCD(10,2) ==> $r = 10 \% 2 ==> 0$

$m = 2$ $n = 0$

GCD = $m = 2$

GCD(10,15) ==> $r = 10 \% 15 ==> 10$

$m = 15$ $n = 10$

==> $r = 15 \% 10 ==> 5$

$m = 10$ $n = 5$

==> $r = 10 \% 5 ==> 0$

$m = 5$ $n = 0$

GCD = 5

How to design an algorithm?

1- Choose appropriate data structures (e.g., arrays, queues, stacks, trees, graphs, ...etc)

to represent the input and the output of the algorithm.

2- Choose an appropriate algorithm design technique (e.g., dynamic programming) to produce the output from the input.

3. Write the steps of your algorithm.