Computer Science Department
College of Computer and Information Sciences
King Saud University

CSC 311: Design and Analysis of Algorithms[1]
Dr. Waleed Alsalih

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.

[1] This is a summary of the material we cover from the textbook: *Introduction to the Design & Analysis of Algorithms*, A. Levitin, Second Edition, Pearson Addison-Wesley, 2006.

Algorithm 1: Finding the greatest common divisor.

```
Algorithm GCD($m$,$n$)
while $n \neq 0$ do
    $r$:=$m$ mod $n$;
    $m$:=$n$;
    $n$:=$r$;
end
return $m$;
```

How to design an algorithm to solve a problem?

1. Understand the problem and analyze it.
2. Choose appropriate data structures (e.g., arrays, queues, stacks, trees, graphs, ...etc) to represent the input and the output of the algorithm.
3. Think and find a plan to solve the problem (i.e., a sequence of steps to produce the output from the input). It will help to choose an appropriate algorithm design technique (e.g., dynamic programming).
4. Write the steps of your algorithm to solve the problem.