COURSE NAME: ALGORITHMS
COURSE CODE: CIS 212

SYED TANGIM PASHA

LECTURER,

DEPARTMENT OF COMPUTING AND INFORMATION SYSTEM (CIS)

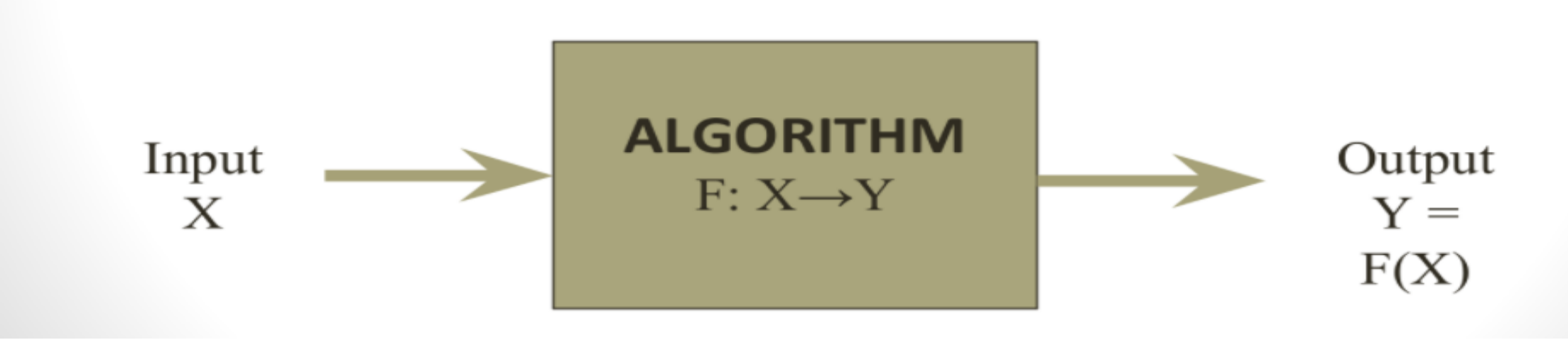
DAFFODIL INTERNATIONAL UNIVERSITY (DIU)

DHAKA,BANGLADESH

What is an Algorithm?

An algorithm is a step by step procedure for solving a particular problem in a finite amount of time.

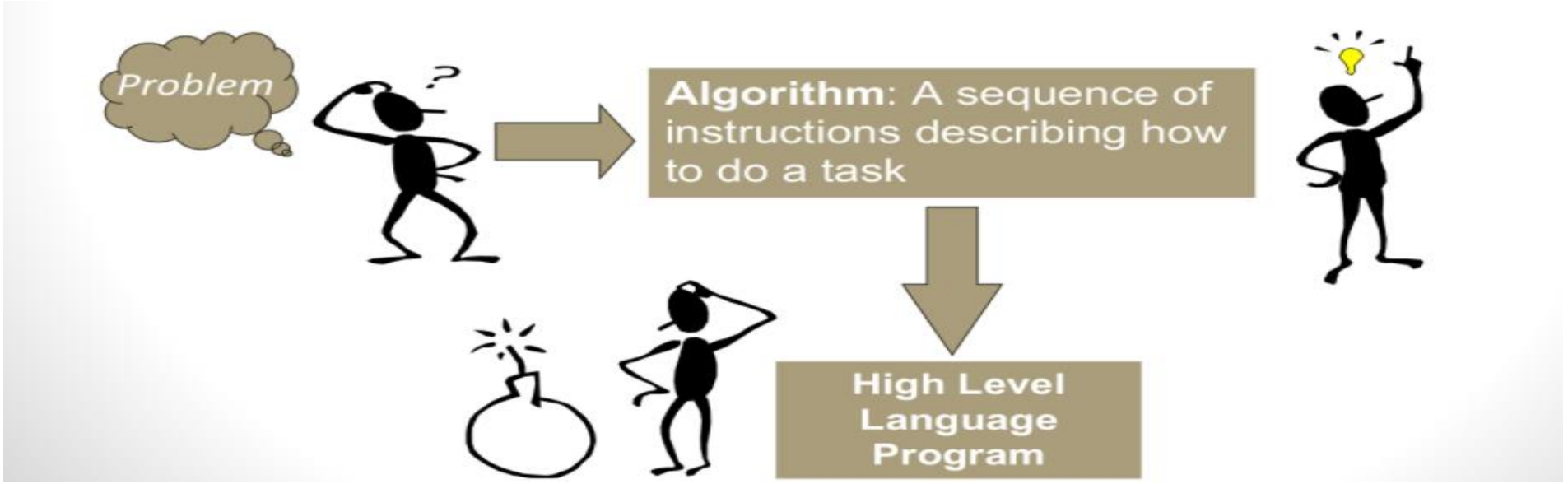
More generally, an algorithm is any well defined computational procedure that takes collection of elements as input and produces a collection of elements as output.



Algorithm vs Program

- A computer program is an instance, or concrete representation, for an algorithm in some programming language.

- Set of instructions which the computer follows to solve a problem.



Brief History

- The study of algorithms began with mathematicians and was a significant area of work in the early years. The goal of those early studies was to find a single, general algorithm that could solve all problems of a single type.
- Named after 9th century Persian Muslim mathematician **Abu Abdullah Muhammad Ibn Musa Al-Khwarizmi** who lived in Baghdad and worked at the Dar Al-Hikmah.

- Dar Al-Hikmah acquired & translated books on science & philosophy, particularly those in Greek, as well as publishing original research.
- The word "algorism" originally referred only to the rules of performing arithmetic using Hindu-Arabic numerals, but later evolved to include all definite procedures for solving problems.

Al-Khwarizmi's Golden Principle

All complex problems can be and must be solved using the following simple steps:

1. Break down the problem into small, simple sub-problems.
2. Arrange the sub-problems in such an order that each of them can be solved without effecting any other.
3. Solve them separately, in the correct order.
4. Combine the solutions of the sub-problems to form the solution of the original problem.

Types of Algorithms

- **Brute Force:** Try all possible solutions until a satisfactory solution is found.
- **Divide and Conquer:** Divide the problem into smaller sub problems of the same type, solve those smaller problems, and combine those solutions to solve the original problem. Ex: Merge Sort algorithm
- **Decrease and Conquer:** Reduce problem instance to smaller instance of the same problem and extend solution. Ex: Binary Search

Types of Algorithms

- **Greedy algorithms:** find an optimal solution at the local level with the intent of finding an optimal solution for the whole problem. Ex: Fractional Knapsack Problem

- **Backtracking algorithms:** Divide the problem into sub problems, each which can be attempted to be solved, however if the desired solution is not reached, move backwards in the problem until a path is found that moves it forward. Ex: Sudoku solving problem

- **Dynamic Programming:** Break a complex problem into a collection of simple sub problems, then solve each of those sub problems only once, storing their solution for future use instead of re-comuputing their solutions. Ex: Fibonacci Series

Algorithm Specification

- **Inputs:** If an algorithm says to take inputs, it should be well-defined inputs.
- **Outputs:** The algorithm must clearly define what output will be yielded and it should be well-defined as well.
- **Finiteness:** If we trace out the instructions of an algorithm, then for all cases, the algorithm terminates after a finite number of steps.
- **Definiteness:** Each instruction is clear and unambiguous.
- **Feasible:** The algorithm must be simple, generic and practical, such that it can be executed upon with the available resources. It must not contain some future technology, or anything.
- **Language Independent:** The algorithm designed must be language independent, it must be just plain instructions that can be implemented in any language, and yet the output will be same, as expected.