Introduction

Algorithm

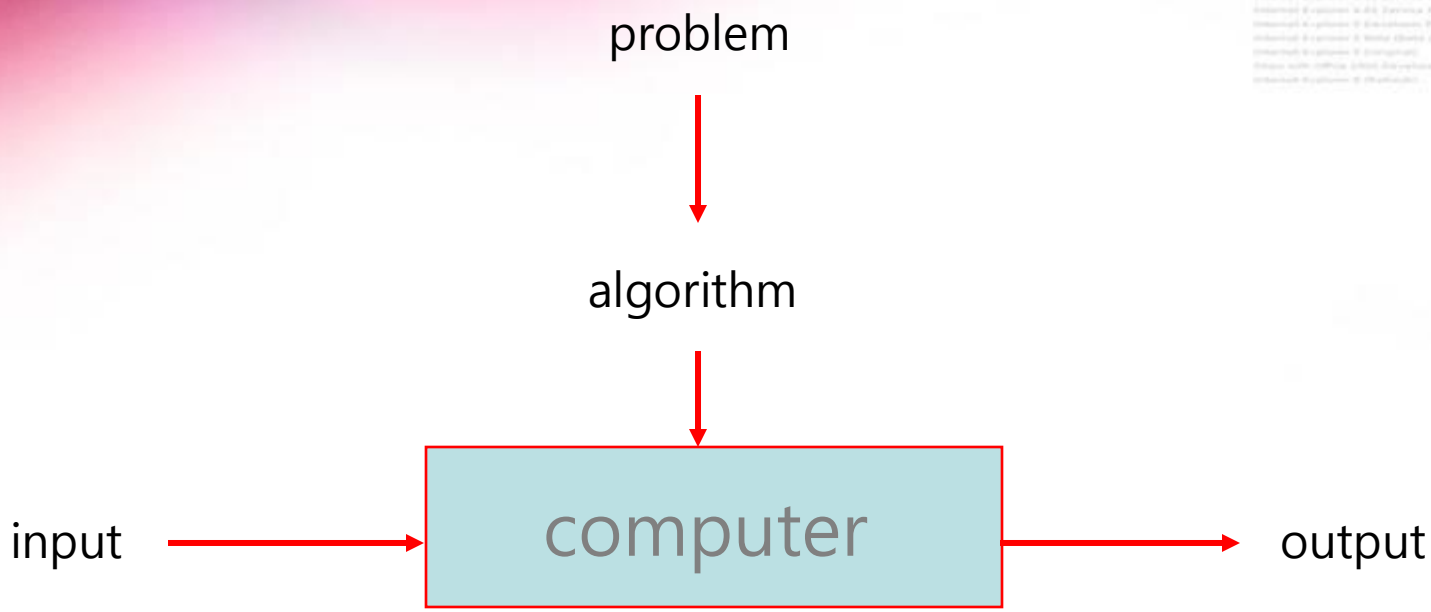
2014 Fall Semester

SUNG KYUN KWAN UNIVERSITY

Algorithm

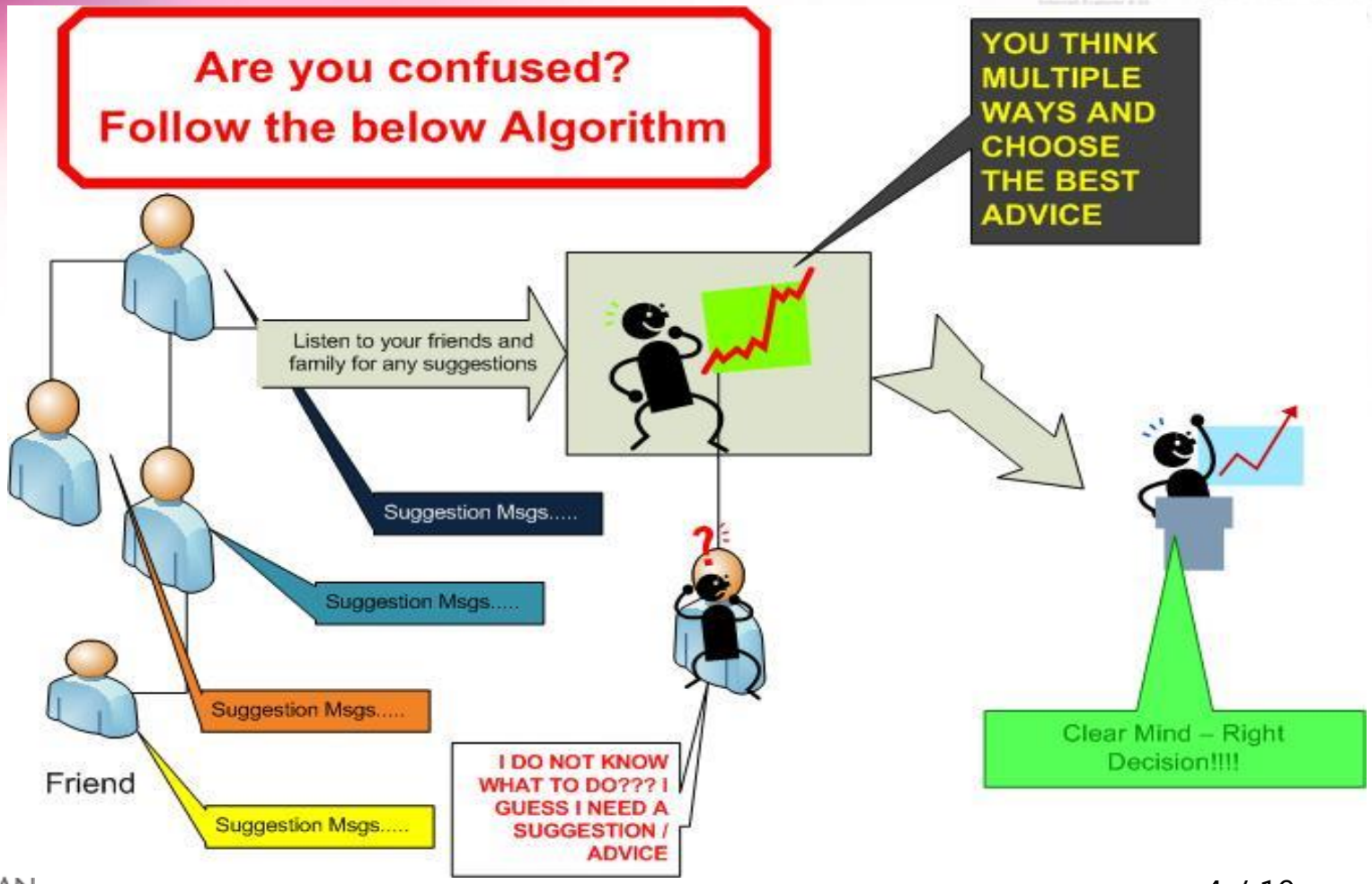
- An *algorithm* is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.

Notion of algorithm



Algorithmic solution

Algorithm in your life



Some Well-known Computational Problems

- Sorting
- Searching
- Shortest paths in a graph
- Minimum spanning tree
- Traveling salesman problem
- Knapsack problem
- Chess
- Towers of Hanoi
- Program termination

Basic Issues Related to Algorithms

- How to design algorithms
- How to express algorithms
- Proving correctness
- Efficiency
- Optimality

Analysis of Algorithms

- How good is the algorithm?
  - Correctness
  - Time efficiency
  - Space efficiency
- Does there exist a better algorithm?

What is an algorithm?

Requirements:

1. Finiteness
   - terminates after a finite number of steps
2. Definiteness
   - unambiguously specified
3. Input
   - valid inputs are clearly specified
4. Output
   - can be proved to produce the correct output given a valid input
5. Effectiveness
   - steps are sufficiently simple and basic

Why study algorithms?

- Theoretical importance
  - the core of computer science
- Practical importance
  - Framework for designing and analyzing algorithms for new problems

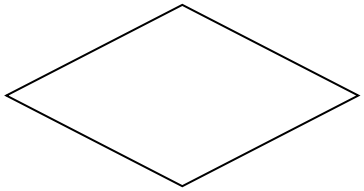
Express Algorithms

- Expressing algorithms
  - Natural languages
  - Pseudo code
  - Flowcharts
  - Programming languages

Flowchart Symbols

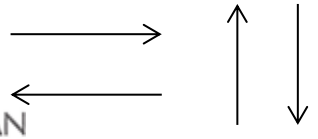
Begin / End

Action step (Processing)



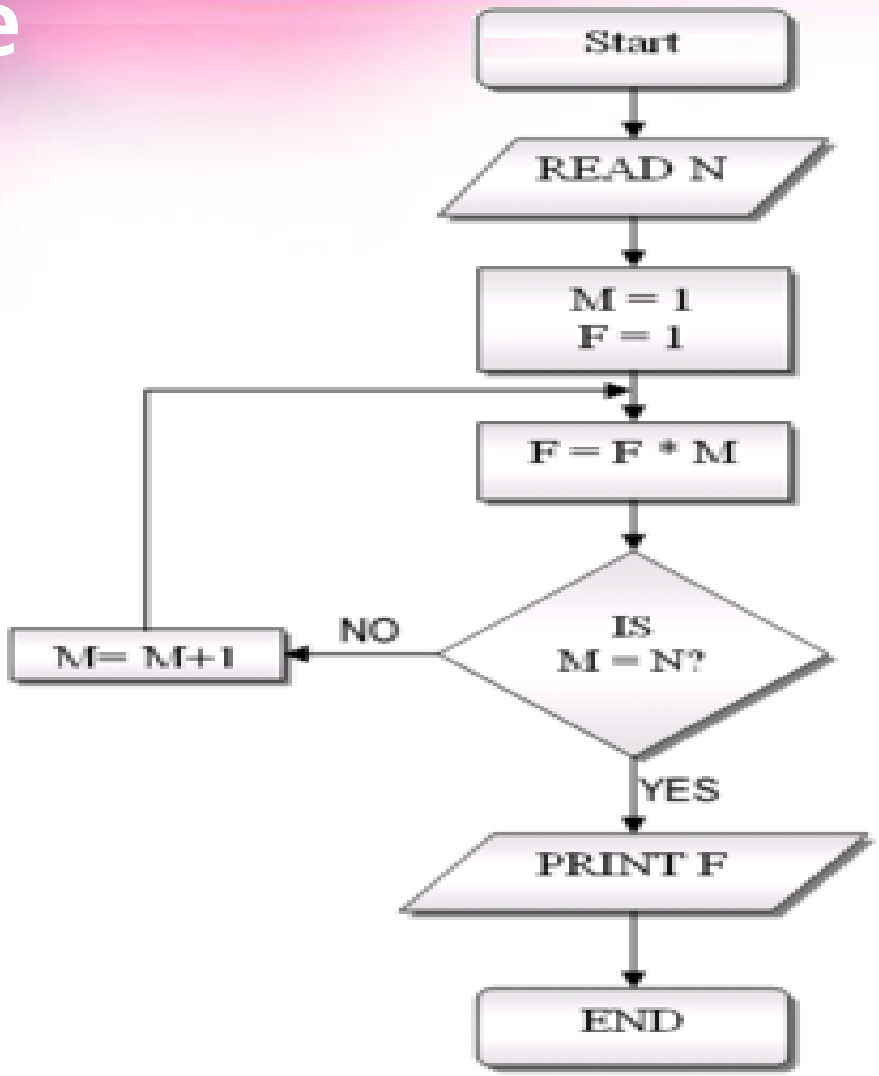
Decision making

Input / Output



Flow Lines

Simple Flowchart Example



SUNG KYUN KWAN UNIVERSITY

Two main issues related to algorithms

- How to design algorithms
- How to analyze algorithm efficiency

Algorithm design techniques/strategies

- Brute force
- Divide and conquer
- Decrease and conquer
- Transform and conquer
- Greedy Method
- Dynamic Programming
- Backtracking

Important problem types

- sorting
- searching
- string processing
- graph problems
- combinatorial problems
- geometric problems
- numerical problems

Problems of techniques (1)

- Brute Force
  - Selection Sort
  - Bubble Sort
  - String Matching
  - Sequential Search
  - Traveling Salesman Problem
  - Knapsack Problem
  - Job Assignment Problem
  - Hamilton Circuits
- Divide and Conquer
  - Merge Sort
  - Quick Sort
  - Binary Search
  - Closest Pair
  - Strassen's Matrix Multiplication
  - Convex Hull

Problems of techniques (2)

- Decrease and Conquer
  - Insertion Sort
  - Graph Traversal Algorithm : Depth First Search
  - Graph Traversal Algorithm : Breadth First Search
  - Topological Sorting
  - Fake coin Problems
  - Multiplication à la russe
  - Josephus Problem
  - Euclid's Algorithm
- Transform and Conquer
  - Presorting
  - Gaussian Elimination
  - Heap Sort

Problems of techniques (3)

- Greedy Methods
  - Prim's Minimal Spanning Tree Algorithm
  - Kruskal's Minimal Spanning Tree Algorithm
  - Dijkstra's Single-Source Shortest Paths Algorithm
  - Coin Change
  - Scheduling Problem
  - Connecting Wires
  - Collecting Gold Coins
- Dynamic Programming
  - Floyd's Algorithm - Shortest Paths
  - Warshall's Algorithm - Shortest Paths
  - 0/1 Knapsack Problem
  - Optimal Binary Search Trees

Problems of techniques (4)

- Backtracking
  - Solving a Maze
  - Coloring a Map
  - Traveling salesperson
  - The Queens Problem
  - Branch and Bound - Assignment Problem

Q & A

SUNG KYUN KWAN UNIVERSITY