TK3043 Analysis and Design of Algorithms

Introduction to Algorithms

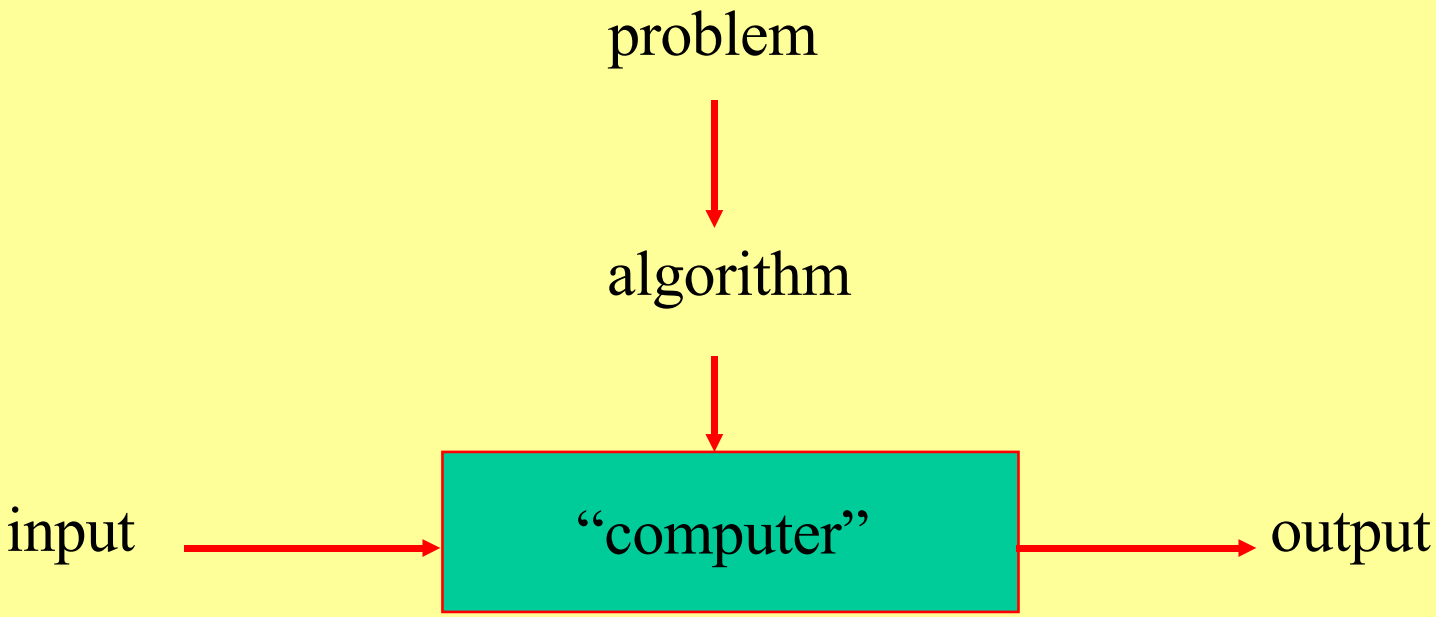
WHAT IS AN ALGORITHM?

- 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.

HISTORICAL PERSPECTIVE

- Euclid's algorithm for finding the greatest common divisor
  - One of the oldest algorithms known (300 BC).
- Muhammad ibn Musa al-Khwarizmi – 9th century mathematician
  - The Al-Khwarizmi principle states that all complex problems of science must be and can be solved by means of five simple steps.

NOTION OF ALGORITHM



Algorithmic solution

EXAMPLE: SORTING

- Statement of problem:
  - Input:

    A sequence of n numbers

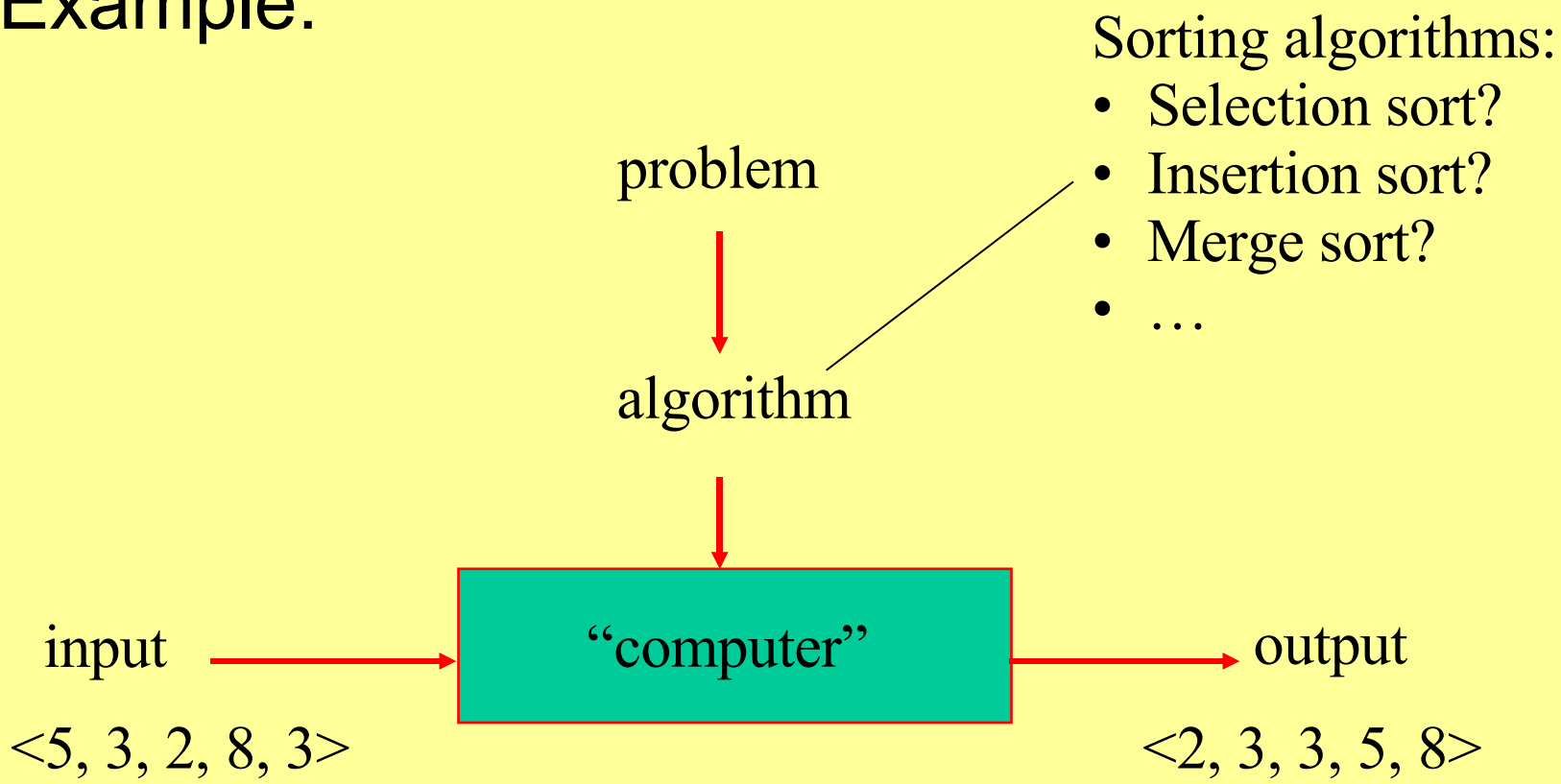
    $\langle a_1, a_2, \ldots, a_n \rangle$
  - Output:

    A reordering of the input sequence

    $\langle b_1, b_2, \ldots, b_n \rangle$

    so that $b_i \leq b_j$ whenever $i < j$

- Example:


Sorting algorithms:
• Selection sort?
• Insertion sort?
• Merge sort?
• …
problem
algorithm
input
“computer”
output
<5, 3, 2, 8, 3>
<2, 3, 3, 5, 8>

SELECTION SORT

- Input:

  array `a[0], …, a[n-1]`

- Output:

  array `a` sorted in non-decreasing order

- Algorithm:

```
for i=0 to n-1
 swap a[i] with smallest of a[i],…a[n-1]
```

SOME WELL-KNOWN COMPUTATIONAL PROBLEMS

- Sorting
- Searching
- Shortest paths in a graph
- Minimum spanning tree
- Primality testing
- 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 of algorithms
- Efficiency
  - Theoretical analysis
  - Empirical analysis
- Optimality

ALGORITHM DESIGN STRATEGIES

- Brute force
- Divide and conquer
- Decrease and conquer
- Transform and conquer
- Greedy approach
- Dynamic programming
- Backtracking and Branch and bound
- Space and time tradeoffs

ANALYSIS OF ALGORITHMS

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

WHAT IS AN ALGORITHM?

- Recipe, process, method, technique, procedure, routine,… with following requirements:
  1. Finiteness
     terminates after a finite number of steps
  2. Definiteness
     rigorously and 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
  - A practitioner's toolkit of known algorithms
  - Framework for designing and analyzing algorithms for new problems

EUCLID'S ALGORITHM

- Problem:

  Find gcd(m,n), the greatest common divisor of two nonnegative, not both zero integers m and n

- Examples:

  gcd(60,24) = 12

  gcd(60,0) = 60

  gcd(0,0) = ?

EUCLID'S ALGORITHM

- Euclid's algorithm is based on repeated application of equality

  $$\gcd(m,n) = \gcd(n, m \bmod n)$$

  until the second number becomes 0, which makes the problem trivial.
- Example:

  $$\gcd(60,24) = \gcd(24,12) = \gcd(12,0) = 12$$

TWO DESCRIPTIONS OF EUCLID'S ALGORITHM

- Step 1
  If n = 0, return m and stop; otherwise go to Step 2
- Step 2
  Divide m by n and assign the value fo the remainder to r
- Step 3
  Assign the value of n to m and the value of r to n. Go to Step 1.

```
while n ≠ 0 {
    r ← m mod n
    m← n
    n ← r
}
return m
```

OTHER METHODS FOR COMPUTING gcd(m,n)

- Consecutive integer checking algorithm
  - 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

OTHER METHODS FOR COMPUTING gcd(m,n) (CONT.)

- Middle-school procedure
  - 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)

  Is this an algorithm?

IMPORTANT PROBLEM TYPES

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

FUNDAMENTAL DATA STRUCTURES

- list
  - array
  - linked list
  - string
- stack
- queue
- priority queue
- graph
- tree
- set and dictionary