MA144: Problem Solving and Computer Programming

Lecture-3

Algorithm, Pseudocode

Problem

- Any real world situation which can be expressed unambiguously and that has a solution or procedure to solve it.

- Problems are not restricted to mathematical problems

- Recipes found in cookery books are procedures (or solutions) for solving a cookery problem.
  - How to prepare a coffee?

Problem Solving

- Problem Solving is the sequential process of analyzing information related to a given situation and generating appropriate response options.
- The following 6 steps must be followed to solve a problem:
  1. Understand the Problem
  2. Formulate a Model
  3. Develop an Algorithm
  4. Write a Program (in this course, C++)
  5. Test the Program
  6. Evaluate the Solution

Understand the Problem

- What input data/information is available?
- What does it represent?
- What format it is in?
- Is anything missing?
- What output information I am trying to produce?
- What I am going to have to compute
- What is the format of the output, like text, picture, graph …

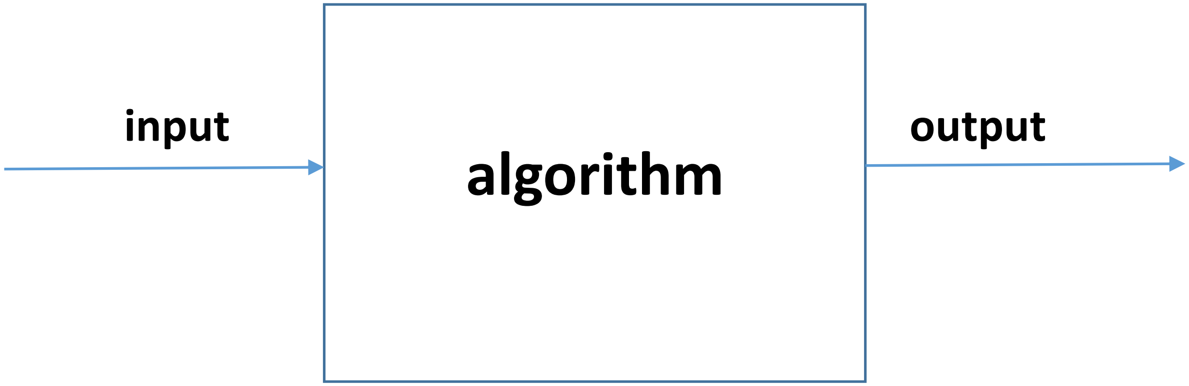
Formulate a Model

- Identify a formula or a mathematical expression for solving the problem

- Model, to solve the problem, can be developed after analyzing the problem and its possible solution.
  - In the problem of finding the sum,
    the values are numbers and hence can be added
  - if they are not numbers but grades with different credits for each course,
    to find cumulative grade point,
    the weighted average can be calculated

Algorithm

- **Algorithm:** A sequence of simple, unambiguous and effectively computable statements, that when executed sequentially, produces the desired result and halts in a finite amount of time.
  - An algorithm is a sequence of computational steps that transforms the input into the output.



Characteristics of an Algorithm

Simple: A statement that is written in a simple language and is understandable to the reader.

- Complex and compound statements are to be avoided.

Unambiguous: A statement without ambiguities and is based on mathematical facts

- The expression a/b/c. Is it a/(b/c) or (a/b)/c?

Effectively Computable: A statement that can be properly executed

- x is sum of all positive integers is simple but is not effectively computable

Characteristics of an Algorithm (contd...)

Sequential execution: All the statements must be executed in sequential order one after other – including branching statements

Finite time: Time taken to complete the solution must be finite

- Add 1 to each integer is simple, unambiguous, effictive but can not be executed in finite amount of time.

Structured Programming

Three control structures:

- Sequence
- Selection
- Repetition

Sequence: the construct where one statement is executed after another

Statement 1
Statement 2
Statement 3
⋮

Structured Programming (contd)

Selection: the construct where statements are executed or skipped depending on whether a condition evaluates to TRUE or FALSE

There are three selection structures:

- IF
- IF-ELSE
- SWITCH

Structured Programming (contd)

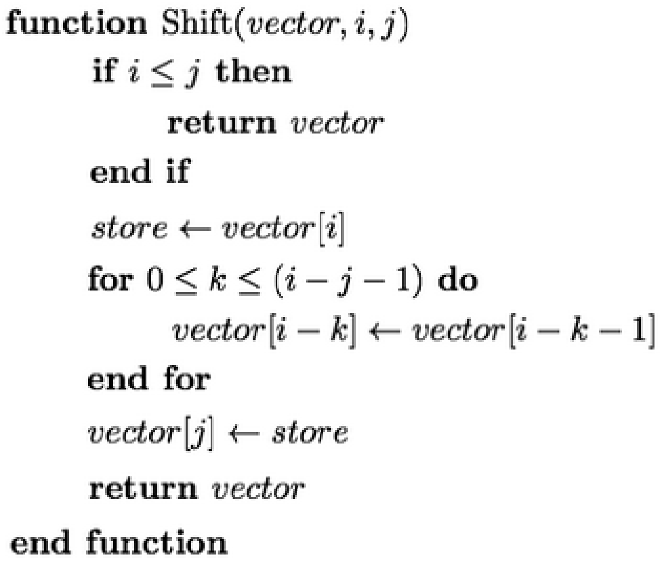
Repetition: the construct where statements can be executed repeatedly until a condition evaluates to TRUE or FALSE

There are three repetition structures

- FOR
- WHILE
- DO-WHILE

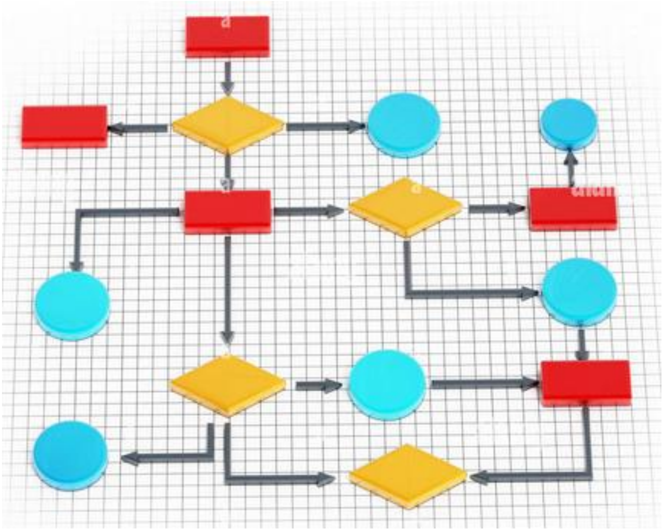
Description of an Algorithm

- **Pseudocode**



```
function Shift($vector, i, j$)
        if $i \leq j$ then
                return $vector$
        end if
        $store \leftarrow vector[i]$
        for $0 \leq k \leq (i - j - 1)$ do
                $vector[i - k] \leftarrow vector[i - k - 1]$
        end for
        $vector[j] \leftarrow store$
        return $vector$
end function
```

- **Flowchart**



Pseudocode (algorithm design language)

- Consists of natural language-like statements that precisely describe the steps of an algorithm
- Statements describe actions
- Focuses on the logic of the algorithm
- Avoids language-specific elements
- Steps are numbered
- Indentation is used for dependent statements in selection and repetition structures

Pseudocode Language Constructs

Assignment/Computation

Assign expression to var1	var1 ← expression
Compute var2 as the sum of x and y	var2 ← $x + y$
Increment counter	counter ←counter+1

Input

Get var1, var2, …	**Read** var1, var2, …

Output

Display var1, var2, …	**Write** var1, var2, …

Pseudocode Language Constructs (contd)

Selection

```
IF condition
    statement1
    statement2
statementA
```

```
IF condition
    statement1
    statement2
ELSE
    statement3
    statement4
statementA
```

```
SWITCH expression
    case1: action1
    case2: action2
          ⋮
    default: actionk
statementA
```

Pseudocode Language Constructs (contd)

Repetition

```
FOR bounds on repetition
        satement1
        satement2
statementA
```

```
WHILE  condition
     statement1
     statement2
statementA
```

```
DO
     statement1
     statement2
WHILE  condition
statementA
```

Pseudocode Examples

1. Find the average of given four numbers
2. Find a profit or loss
3. Print a multiplication table of a given number
4. Calculate factorial of a given number
5. Find the maximum of more than three numbers
6. Exchange the values of two variables
7. Find gcd of two numbers
8. Compute $a^k \bmod n$
9. Check whether the given number is prime or not
10. Locate all the prime numbers between 1 and the given number n
11. Find lcm of two numbers

Next Lecture

Flowcharts