CSE231: Data Structures

Lecture 03

by

Mehedi Hasan Bijoy

Mathematical Notations and Functions

- **Floor and Ceiling Functions**
  - The floor and ceiling functions give us the nearest integer up or down.

$\lfloor x \rfloor$, called the *floor* of x,
$\lceil x \rceil$, called the *ceiling* of x,

Mathematical Notations and Functions

- **Reminder Function (%)**

  $k \pmod M$

  $25 \pmod 7 = 4, \quad 25 \pmod 5 = 0, \quad 35 \pmod{11} = 2, \quad 3 \pmod 8 = 3$

- **Summation**

$$a_1 + a_2 + \cdots + a_n \quad \text{and} \quad a_m + a_{m+1} + \cdots + a_n$$

$$\sum_{j=1}^{n} a_j \quad \text{and} \quad \sum_{j=m}^{n} a_j$$

Mathematical Notations and Functions

- **Factorial**
  The product of the positive integers from 1 to n, inclusive, is denoted by n!

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-2)(n-1)n$$

- **Permutation**
  A permutation of a set of n elements is an arrangement of the elements in a given order. The permutations of the set consisting of the elements a, b, c are as follows:

abc, *acb*, *bac*, *bca*, *cab*, *cba*

Mathematical Notations and Functions

- **Exponents**

$$a^m = a \cdot a \cdots a \ (m \text{ times}), \quad a^0 = 1, \quad a^{-m} = \frac{1}{a^m}$$

$$a^{m/n} = \sqrt[n]{a^m} = \left(\sqrt[n]{a}\right)^m$$

- **Logarithms**

$$y = \log_b x \quad \text{and} \quad b^y = x$$

$\log_2 8 = 3$ since $2^3 = 8$; $\log_{10} 100 = 2$ since $10^2 = 100$

$\log_2 64 = 6$ since $2^6 = 64$; $\log_{10} 0.001 = -3$ since $10^{-3} = 0.001$

$\log_b 1 = 0$ since $b^0 = 1$

$\log_b b = 1$ since $b^1 = b$

Algorithm

- An algorithm is a finite sequence of rigorous instructions, typically used to solve a class of specific problems or to perform a computation.
    - It refers to a finite step-by-step list of well-defined logical instructions for solving a particular problem.

Cup of Tea Algorithm:

Set of precise instructions to solve a problem or make something happen.

1. 
Fill kettle with enough water to make a cup of tea.

2. 
Plug kettle in and switch on.

3. 
Place a tea bag in a mug.

4. 
When the kettle has boiled pour water in to the mug to near the top.

5. 
Leave tea bag until tea is strong enough.

6. 
With a spoon take out the tea bag.

7. 
Add milk to the tea.

8. 
Once the tea is cool enough, drink.

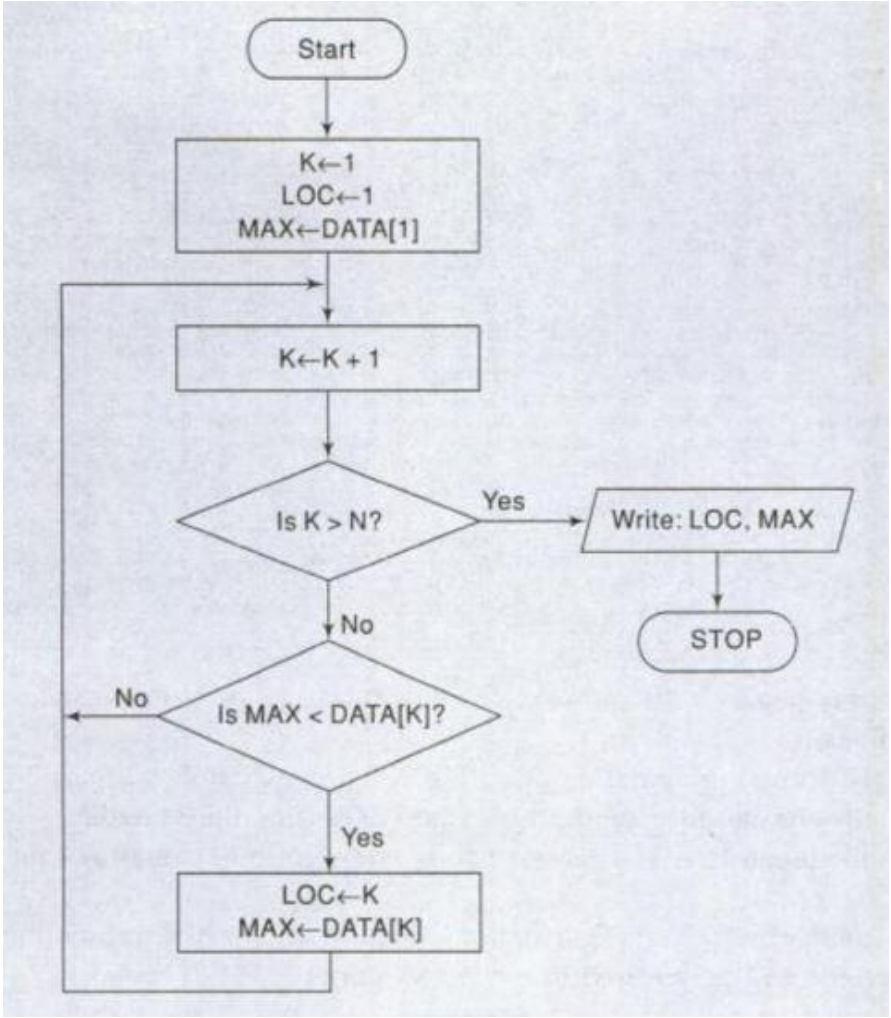
An array DATA of numerical values is in memory. We want to find the location LOC and the value MAX of the largest element of DATA. Given no other information about DATA, one way to solve the problem is as follows:

Initially begin with LOC = 1 and MAX = DATA[1]. Then compare MAX with each successive element DATA[K] of DATA. If DATA[K] exceeds MAX, then update LOC and MAX so that LOC = K and MAX = DATA[K]. The final values appearing in LOC and MAX give the location and value of the largest element of DATA.

A formal presentation of this algorithm, whose flow chart appears in Fig. 2.2, follows.

Algorithm 2.1: (Largest Element in Array) A nonempty array DATA with N numerical values is given. This algorithm finds the location LOC and the value MAX of the largest element of DATA. The variable K is used as a counter.

Step 1. [Initialize.] Set K := 1, LOC := 1 and MAX := DATA[1].
Step 2. [Increment counter.] Set K := K + 1.
Step 3. [Test counter.] If K > N, then:
Write: LOC, MAX, and Exit.
Step 4. [Compare and update.] If MAX < DATA[K], then:
Set LOC := K and MAX := DATA[K].
Step 5. [Repeat loop.] Go to Step 2.


Start
K←1
LOC←1
MAX←DATA[1]
K←K + 1
Is K > N?
Yes
Write: LOC, MAX
STOP
No
Is MAX < DATA[K]?
No
Yes
LOC←K
MAX←DATA[K]

Control Structures

- Algorithms and their equivalent computer programs are more easily understood if they mainly use self-contained modules and three types of logic, or flow of control, called
  - Sequence logic / sequential flow
  - Selection logic / conditional flow
  - Iteration logic / repetitive flow

Sequence Logic / Sequential Flow

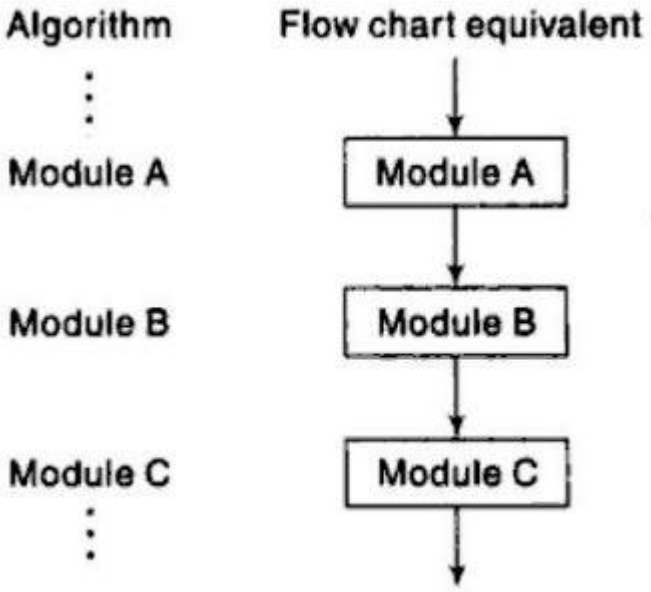


Fig. 2.3 Sequence Logic

Selection Logic / Conditional Flow

- Single Alternative

```
If condition, then:
    [Module A]
[End of If structure.]
```

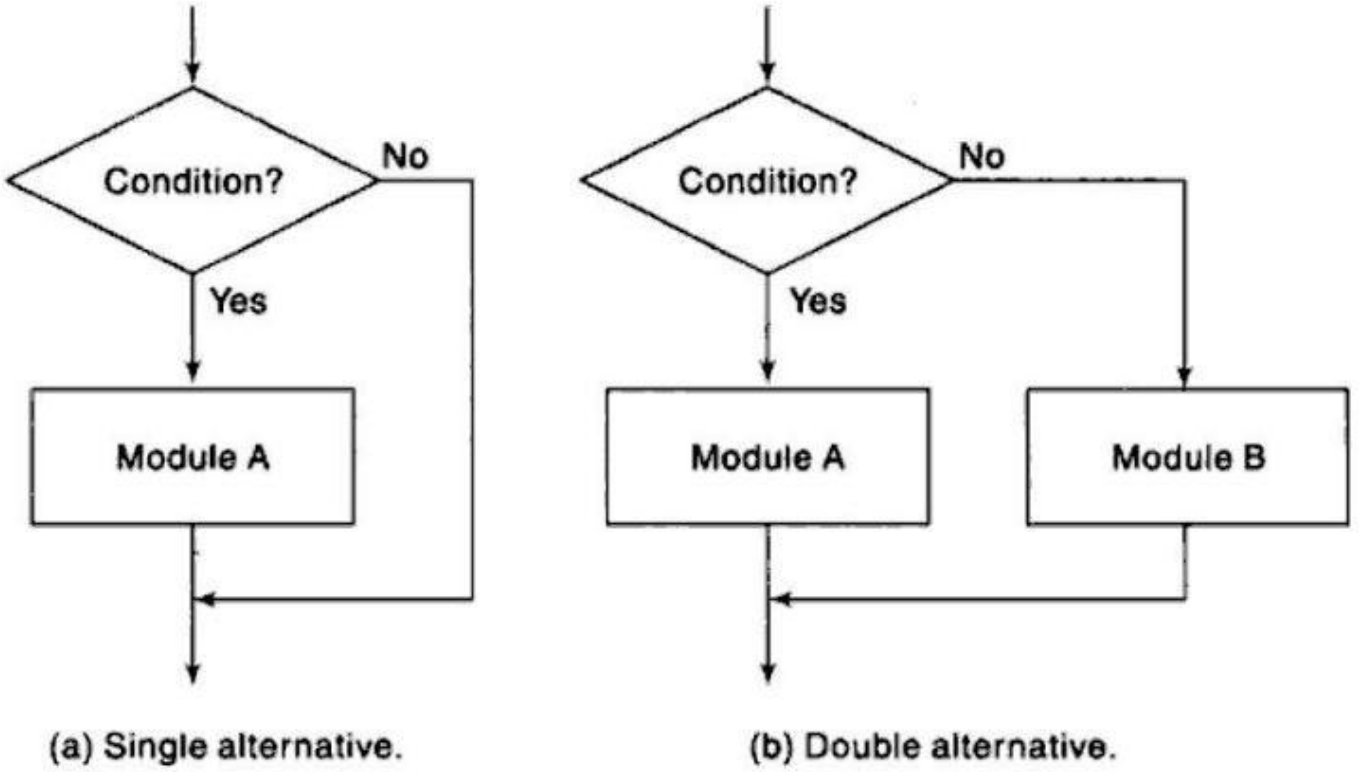
- Double Alternative

```
If condition, then:
    [Module A]
Else:
    [Module B]
[End of If structure.]
```

- Multiple Alternative

```
If condition(1), then:
    [Module A_1]
Else if condition(2), then:
    [Module A_2]
    ⋮
Else if condition(M), then:
    [Module A_M]
Else:
    [Module B]
[End of If structure.]
```

Selection Logic / Conditional Flow



(a) Single alternative.

(b) Double alternative.

The solutions of the quadratic equation

$$ax^2 + bx + c = 0$$

where $a \neq 0$, are given by the quadratic formula

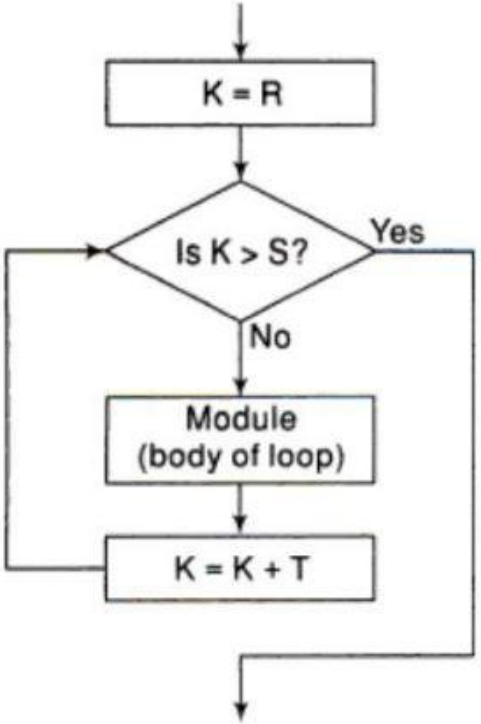
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The quantity $D = b^2 - 4ac$ is called the *discriminant* of the equation. If D is negative, then there are no real solutions. If $D = 0$, then there is only one (double) real solution, $x = -b/2a$. If D is positive, the formula gives the two distinct real solutions. The following algorithm finds the solutions of a quadratic equation.

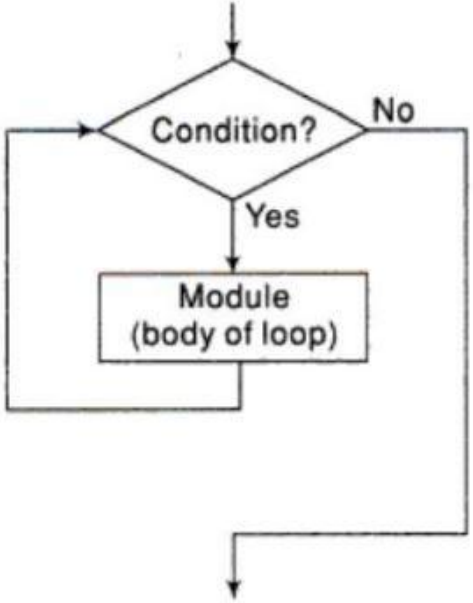
Algorithm 2.2: (Quadratic Equation) This algorithm inputs the coefficients A, B, C of a quadratic equation and outputs the real solutions, if any.

Step 1. Read: A, B, C.
Step 2. Set D: = $B^2 - 4AC$.
Step 3. If D > 0, then:
(a) Set X1 := $(-B + \sqrt{D})/2A$ and
X2 := $(-B - \sqrt{D})/2A$.
(b) Write: X1, X2.
Else if D = 0, then:
(a) Set X := –B/2A.
(b) Write: 'UNIQUE SOLUTION', X.
Else:
Write: 'NO REAL SOLUTIONS'
[End of If structure.]
Step 4. Exit.

Iteration Logic / Repetitive Flow



(a) Repeat-For structure.



(a) Repeat-While structure.

Iteration Logic / Repetitive Flow

(Largest Element in Array) Given a nonempty array DATA with N numerical values, this algorithm finds the location LOC and the value MAX of the largest element of DATA.

1. [Initialize.] Set K := 1, LOC := 1 and MAX := DATA[1].
2. Repeat Steps 3 and 4 while K ≤ N:
3. If MAX < DATA[K], then:
   Set LOC := K and MAX := DATA[K].
   [End of If structure.]
4. Set K := K + 1.
   [End of Step 2 loop.]
5. Write: LOC, MAX.
6. Exit.

Thank You!