Welcome to the Class

Daffodil International University

DIU

Department of Computing and Information System

Data Structure

Md. Selim Hossain
Senior Lecturer
Department of Computing and Information System
Daffodil International University (DIU),Dhaka, Bangladesh

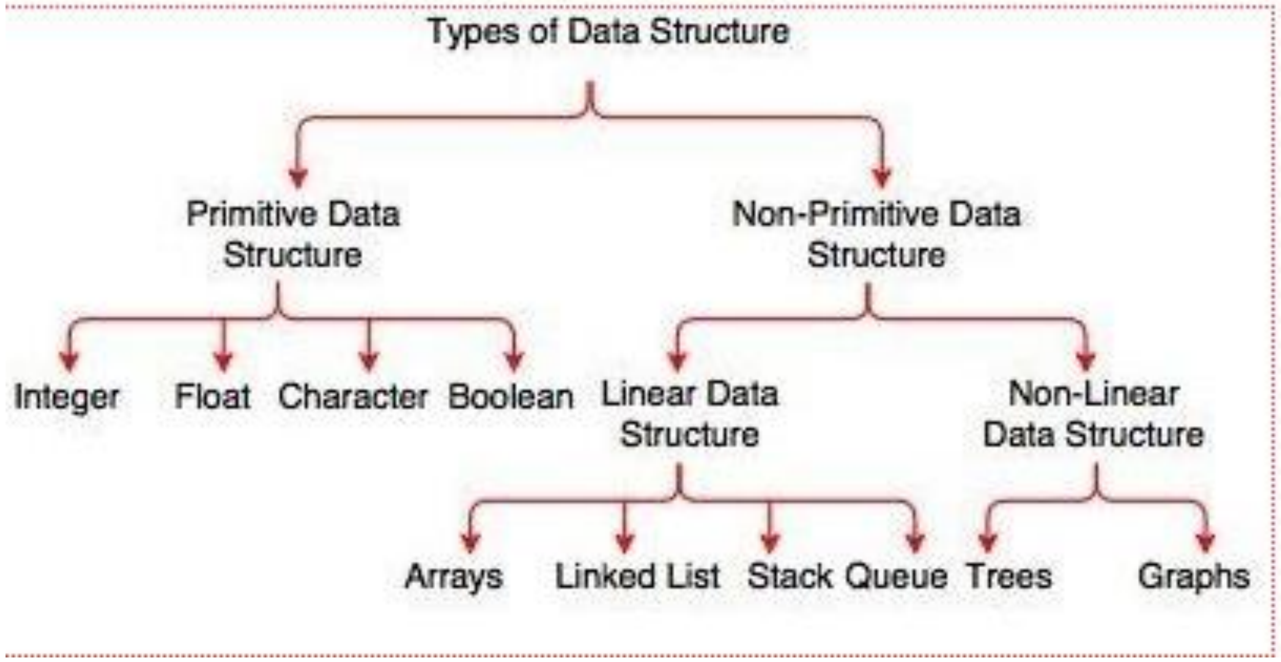
What is Data Structure?

- A Data Structure can be defined informally as an organized collection of values and a set of operations on them.

- Three Components:
  - A set of function definitions
  - A storage structure
  - A set of algorithms

Data structures can also be classified on the basis of the following characteristics

Characterstic	**Description**
Linear	In Linear data structures,the data items are arranged in a linear sequence. Example: **Array**
Non-Linear	In Non-Linear data structures,the data items are not in sequence. Example: **Tree**, **Graph**
Homogeneous	In homogeneous data structures,all the elements are of same type. Example: **Array**
Non-Homogeneous	In Non-Homogeneous data structure, the elements may or may not be of the same type. Example: **Structures**
Static	Static data structures are those whose sizes and structures associated memory locations are fixed, at compile time. Example: **Array**
Dynamic	Dynamic structures are those which expands or shrinks depending upon the program need and its execution. Also, their associated memory locations changes. Example: **Linked List created using pointers**


Types of Data Structure
Primitive Data Structure
Non-Primitive Data Structure
Integer
Float
Character
Boolean
Linear Data Structure
Non-Linear Data Structure
Arrays
Linked List
Stack
Queue
Trees
Graphs

Importance of Data Structure

- **Data structure** provides the right way to organize information in the digital space.
- The **data structure** is a key component of Computer Science and is largely used in the areas of Artificial Intelligence, operating systems, graphics, etc.

Need for Data Structure

- **Data Search −** Consider an inventory of 1 million(10^6) items of a store. If the application is to search an item, it has to search an item in 1 million(10^6) items every time slowing down the search. As data grows, search will become slower.
- **Processor speed −** Processor speed although being very high, falls limited if the data grows to billion records.
- **Multiple requests −** As thousands of users can search data simultaneously on a web server, even the fast server fails while searching the data.

Characteristics of a Data Structure

- **Correctness** − Data structure implementation should implement its interface correctly.
- **Time Complexity** − Running time or the execution time of operations of data structure must be as small as possible.
- **Space Complexity** − Memory usage of a data structure operation should be as little as possible

Applications.

- **Arrays**.
- **Stacks**.
- **Queues**.
- **Linked Lists**.
- **Trees**.
- Graphs.
- Tries (they are effectively **trees**, but it's still good to call them out separately).
- **Hash Tables**.

Pseudo-Code

- Not computer programs, but are more structured than usual prose.
- Facilitate the high level analysis of a data structure or algorithm.
- Pseudo code is for human reader, not for a computer.
- To communicate high-level ideas, not low level implementation details.

An example of a Pseudo-code

Procedure ArrayMax(A, n):

Input: An array A storing $n \geq 1$ integers

Output: The maximum element in A

```
currentMax ← A[1]
for i ← 2 to n do
        if currentMax < A[i] then
        currentMax ← A[i]
return currenMax
```

Data Structure Operation

- Four Operations:
  - *Traversing:* Accessing Each record exactly once
  - *Searching:* Finding the location of data/record
  - *Inserting:* Adding a new data/record
  - *Deleting:* Removing a data/record

- Special Operations:
  - *Traversing:* Accessing Each record exactly once
  - *Searching:* Finding the location of data/record

Algorithm

- An algorithm is a finite step-by-step list of well-defined instruction for solving a particular problem.

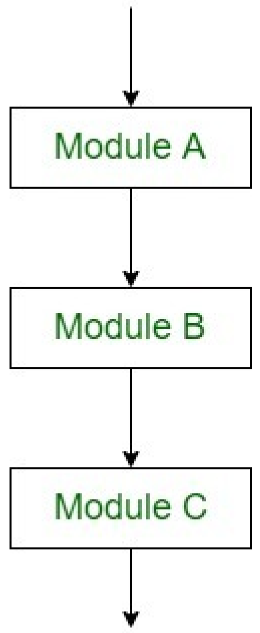
  - **Identifying number**
  - **Step**
  - **Control**
  - **Exit**
  - **Comments**
  - **Variable Name**
  - **Assignment State**
  - **Input / output**

Control structure

- Three types of logic/control logic:
  - *Sequential logic or sequential flow*
  - *Selection logic or conditional flow*
  - *Iteration logic or repetitive flow*

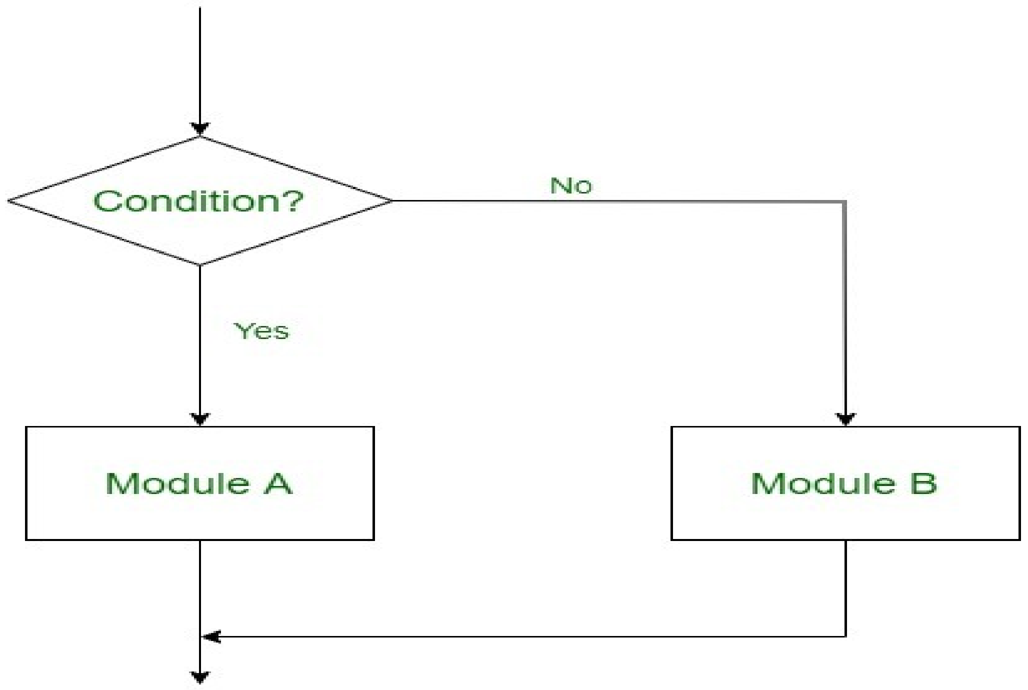
Sequential logic

Sequential logic as the name suggests follows a serial or sequential flow in which the flow depends on the series of instructions given to the computer. Unless new instructions are given, the modules are executed in the obvious sequence.



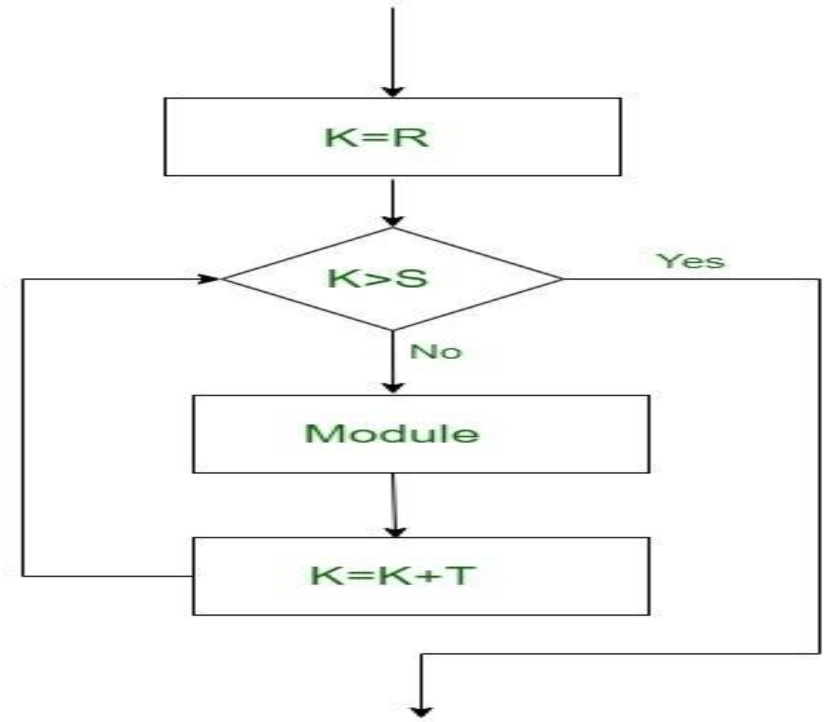
Selection logic

- Selection Logic simply involves a number of conditions or parameters which decides one out of several written modules. The structures which use these type of logic are known as **Conditional Structures**.



Iteration logic

- The Iteration logic employs a loop which involves a repeat statement followed by a module known as the body of a loop.



Step calculations

The step count method is one of the method to analyze the algorithm. In this method, we count number of times one instruction is executing. From that we will try to find the complexity of the algorithm.

Suppose we have one algorithm to perform sequential search. Suppose each instruction will take c1, c2, …. amount of time to execute, then we will try to find out the time complexity of this algorithm

Algorithm	Number of times	Cost
seqSearch(arr, n, key)	1	c1
i := 0	n+1	c2
while i < n, do	n	c3
if arr[i] = key, then	1	c4
break		
end if		
done		
return i	1	c5

Now if we add the cost by multiplying the number of times it is executed, (considering the worst case situation), we will get

$$Cost=c1+(n+1)c2+nc3+c4+c5$$

$$Cost=c1+nc2+c2+nc3+c4+c5$$

$$Cost=n(c2+c3)+c1+c4+c5$$

$$Cost=n(c2+c3)+C$$

Considering the c1 + c4 + c5 is C, so the final equation is like straight line y = mx + b. So we can say that the function is linear. The complexity will be O(n).

- Thanks to All