BASIC DATA STRUCTURES

Learning Outcomes

After the completion of this lesson, the student must be able to:

- understand the importance of data structures and algorithms;
- know the different operations on data structures; and
- implement appropriate data structure in programming;

Data Structures

- programmatic way
- efficient data storage
- code optimization

Importance of Data Structures

- **Data Search**

  - as data grows, search will become slower

- **Processor speed**

  - limited if the data grows to billion records

- **Multiple requests**

  - even the fast server fails while searching the data.

Algorithms

- a step-by-step procedure, which defines
- a set of logical instructions to get the desired output
- created independent of underlying languages

Common Algorithms in Data Structures

- Search
- Sort
- Insert
- Update
- Delete

Sample Applications

- Fibonacci number series
- Knapsack problem
- Tower of Hanoi
- All pair shortest path by Floyd-Warshall
- Shortest path by Dijkstra
- Project scheduling

Features of Data Structures

- **Interface**

  - set of operations and parameters that a data structure supports, accepts, and return type

- **Implementation**

  – internal representation of a data structure

Characteristics of Data Structures

- **Correctness**

  – implementation should implement its interface accurately

- **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

Execution Time Cases

- **Worst Case**

  − This is the scenario where a particular data structure operation takes maximum time it can take. If an operation's worst case time is ƒ(n) then this operation will not take more than ƒ(n) time where ƒ(n) represents function of n

- **Average Case**

  − This is the scenario depicting the average execution time of an operation of a data structure. If an operation takes ƒ(n) time in execution, then m operations will take mƒ(n) time

- **Best Case**

  − This is the scenario depicting the least possible execution time of an operation of a data structure. If an operation takes ƒ(n) time in execution, then the actual operation may take time as the random number which would be maximum as ƒ(n).

Basic Terminologies

- **Data** − set of values
- **Data Item** − single unit of values
- **Group Items** − Data items that are divided into sub items
- **Elementary Items** − Data items that cannot be divided
- **Attribute and Entity** − An entity is that which contains certain attributes or properties, which may be assigned values
- **Entity Set** − Entities of similar attributes form an entity set
- **Field** − single elementary unit of information representing an attribute of an entity
- **Record** − collection of field values of a given entity
- **File** − a collection of records of the entities in a given entity set

Activity # 1

- Using any programming language, write a program implementing the Fibonacci Series

- Make 2 sets of the program using the following specifications:
  1. Using sequential program structure
  2. Using looping with arrays

- Evaluate your code by answering the following questions:
  1. Which of the 2 programs is easier to code?
  2. Which of the 2 programs is more efficient? Why?

Reference/s:

https://www.tutorialspoint.com/data_structures_algorithms/index.htm