RIZAL TECHNOLOGICAL UNIVERSITY

College of Engineering, Architecture and Technology

Boni Campus

Department of Information Technology

Learnings/ Reflection

From

Module 1 to 8

Submitted by:

Reymar Blangquisco

Submitted to:

Prof. Joy Salazar

6 December 2022

Module 1: DATA STRUCTURE I

- I studied about stacks, queues, and their types in Data Structure 1 after the presentation. A stack, from what I recall, is a linear data structure in which all insertion and deletion of data, or its values, is done at one end alone, rather than in the center. A queue is also a linear data structure that is open at both ends and performs operations in First In First Out (FIFO) order. A queue is defined as a list in which all additions to the list occur at one end and all deletions occur at the other.

Module 2: DATA STRUCTURE II

-This module is what we reported and we review this topic thoroughly, some of the topic are hard to understand but thankfully we survive and present it successfully. While doing a review in this topic I've learned that patient and determination to learn is sometimes a key to make those hard to learn into easy to learn.

Module 3: TREE BASED DSA I

-Tree Based DSA 1, has parent nodes and a child node. It is a non-linear data structure and a hierarchy made up of nodes, with each node storing a value and a list of references to other nodes.

Module 4: TREE BASED DSA II

-It's a little bit complicated topic because I see it a broad one. It tackled here the B tree which known as self-balancing search tree that say's each node contains more than one key and can have more than two children, B tree also generalized form of binary search tree.

Module 5: GRAPH BASED DSA

- According to what I learned during their presentation in module 5, a graph data structure is a collection of nodes that contain data and are connected to other nodes.

Module 6: SORTING AND SEARCHING FOR ALGORITHM

-In this module I've learned how to sort elements and putting it in organized order. It's like array but the rules in sorting is kinda different.

Module 7: GREEDY ALGORITHM

- A greedy algorithm is a method of solving problems that selects the best alternative available at the time. It is unconcerned with whether the current best outcome will result in the ultimate best result. Even if the previous judgment was incorrect, the algorithm never reverses it. It operates from the top down. This approach may not provide the optimal solution for all issues. It's because it always looks for the greatest local option to achieve the best global outcome.

However, we can tell if the method is applicable to any situation if it contains the following properties:

1. Greedy Property Selection

A greedy strategy can be used to solve a problem if an optimal solution can be obtained by selecting the best option at each stage without reviewing the previous steps once chosen. This is known as greedy choice property.

2. Substructure Optimization

If the best overall solution to the problem corresponds to the best solution to its subproblems, then the problem can be solved greedily. This is known as optimal substructure.

Module 8: DYNAMIC PROGRAMMING

- Dynamic Programming is a computer programming approach that aids in the effective solution of a class of problems with overlapping subproblems and optimum substructure. If a problem can be broken into subproblems, which are then divided into smaller subproblems, and if these subproblems overlap, the answers to these subproblems can be kept for future reference. The CPU's efficiency can be improved in this manner. This form of problem resolution is known as dynamic programming. To identify the best solution, such issues need continually computing the value of the same subproblems.