Reflection on Data Structures Assignment 1

Name: Bikey Bhujel

ID: C0913677

The Array Artifact

Reflection:
In this challenge, I learned how to use an array to store a list of artifacts. It was interesting to see how arrays work for holding fixed-size collections. I also practiced adding and removing items, which helped me understand basic array operations.

Difficulties:
One issue I faced was managing empty spots when I removed an artifact. I fixed this by shifting the remaining items to fill the gap.

Improvements:
In the future, I could make the array grow when needed, allowing it to hold more artifacts without being limited in size.

The Linked List Labyrinth

Reflection:
This challenge taught me about linked lists. I learned how they allow adding and removing locations easily without shifting others. Keeping track of pointers to the next location was also important for navigating the list.

Difficulties:
Detecting loops in the path was tough at first. I solved this by using a slow and fast pointer method, which worked well.

Improvements:
I could enhance this by adding features like saving the path or showing it visually, making it easier to understand.

The Stack of Ancient Texts

Reflection:
In this challenge, I implemented a stack to manage ancient scrolls. I learned about the Last In, First Out (LIFO) principle, which is useful in various situations, like managing function calls.

Difficulties:
I had a hard time ensuring the correct order of operations when adding and removing scrolls. I managed this by keeping track of the top index.

Improvements:
I could extend this by allowing the stack to resize when needed or by sorting scrolls based on title or age.

The Queue of Explorers

Reflection:
This challenge helped me understand queues, especially circular queues. I learned how they efficiently handle waiting lines for explorers.

Difficulties:
Managing the circular behavior was tricky at first. I used modulo operations to ensure the pointers stayed within the limits.

Improvements:
I could improve this by adding priority levels for explorers, allowing some to enter the temple faster based on certain criteria.

The Binary Tree of Clues

Reflection:
Creating a binary tree of clues was a great learning experience. I learned how binary search trees work and how they allow for quick searching and adding clues.

Difficulties:
Keeping the tree balanced during insertion was challenging. I simplified this by following comparison rules for placing clues correctly.

Improvements:
In the future, I could implement balancing methods to keep the tree efficient, ensuring quick access to clues.