Task_8

Compare between the above data structure types from the

following points:

- ➢ Access approach (eg. FIFO)
- ➢ Time complexity in deletion, insertion and search.
- ➢ Space complexity

Arrays

- Access approach: Arrays can be directly accessed using an index.
- Time complexity: Insertion and deletion operations have a time complexity of O(n) due to the need to shift elements. Searching an element in an array requires iterating through all elements, resulting in a time complexity of O(n).
- Space complexity: is O(n)

Linked Lists

- Access approach: Linked lists require traversal from the beginning or end of the list to access an element.
- Time complexity: Insertion and deletion operations are faster than arrays as it only requires updating the pointers, resulting in a time complexity of O(1). Searching an element

in a linked list requires iterating through all elements, resulting in a time complexity of O(n).

- Space complexity: is O(n)

Binary Search Trees

- Access approach: The access approach in a binary search tree is based on the value of the elements. Elements with values that are greater than the root node are stored on the right subtree while those with values that are less than the root node are stored on the left subtree.
- Time complexity: Insertion, deletion, and search operations have a time complexity of O(log n) in a balanced binary search tree. However, in the worst case, the time complexity can be O(n) when the tree is heavily unbalanced.
- Space complexity: is O(n).

Hash Tables

- Access approach: Hash tables use a hash function to map keys to their corresponding values, enabling direct access to elements.
- Time complexity: Insertion, deletion, and search operations have an average time complexity of O(1). However, in the worst case, the time complexity can be O(n).
- Space complexity: is O(n).

Stack

- Access approach: Stacks follow the Last-In-First-Out (LIFO) approach, where the last element added is the first one to be removed.
- Time complexity: Insertion and deletion operations have a time complexity of O(1). Searching an element in a stack requires iterating through all elements, resulting in a time complexity of O(n).

- Space complexity: is O(n).

Queue

- Access approach: Queues follow the First-In-First-Out (FIFO) approach, where the first element added is the first one to be removed.
- Time complexity: Insertion and deletion operations have a time complexity of O(1). Searching an element in a queue requires iterating through all elements, resulting in a time complexity of O(n).
- Space complexity: The space complexity of a queue is O(n).