

Project 2

(a)

1. Arrays: A collection of elements stored at contiguous memory locations, allowing efficient random access.
2. Linked Lists: A data structure consisting of a sequence of elements where each element points to the next one, allowing for dynamic memory allocation and efficient insertion and deletion operations.
3. Stacks: A Last-In-First-Out (LIFO) data structure where elements are inserted and removed from the same end, commonly used in algorithms for parsing expressions, backtracking, and managing function calls.
4. Queues: A First-In-First-Out (FIFO) data structure where elements are inserted at the rear and removed from the front, commonly used in scheduling tasks, breadth-first search, and implementing buffers.
5. Trees: A hierarchical data structure consisting of nodes connected by edges, with a root node at the top and child nodes branching downwards, commonly used in representing hierarchical relationships and searching algorithms.
6. Binary Trees: A tree data structure where each node has at most two children, commonly used in binary search trees and expression trees.
7. Binary Search Trees (BST): A binary tree data structure where the left child is less than the parent, and the right child is greater, facilitating efficient searching, insertion, and deletion operations.
8. Heaps: A binary tree-based data structure where the parent node is either greater than or equal to (max heap) or less than or equal to (min heap) its children, commonly used in priority queues and heap sort algorithms.
9. Hash Tables: A data structure that implements an associative array abstract data type, where keys are mapped to values using a hash function, allowing for constant-time average case access.
10. Graphs: A data structure consisting of vertices (nodes) and edges (connections) between them, used to represent networks, social connections, and various relationships.