

Lab 08

Data Structures

BS DS Fall 2024 Morning/Afternoon

Objective:

Learning and Implementing the Representation of Binary Trees using Array base and implementing some basic operations on them.

ADT for Array Representation

Note: you can add some utility functions as per requirement for the completion of public functions.

```
template<class T>
class BinaryTree
{
private:
    int height;           //represents the maximum possible nodes in (capacity =  $2^{\text{height}} - 1$ ) tree.
    char *nodeStatus;    //It is used to find that whether there is a node on a particular index of array.
    // Note: we are not using the approach of using a sentinel value in 'data'
    //array because in a template based data T could be of any type.

    T *data;             // stores the nodes of the trees
};
```

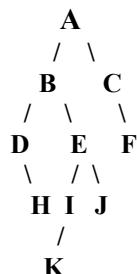
Add the following public methods in the Binary Tree Class;

1. BinaryTree(int h); // initializes nodeStatus array with 0 and creates data array of size $2^h - 1$.
2. void setRoot(T v); //stores v at data[0] as root of tree and also sets the nodeStatus[0] =1.
3. T getRoot(); //returns the root of tree if exists.
4. void setLeftChild(T parent, T child);
5. void setRightChild(T parent, T child);
6. T getParent(T node);
7. void remove(T node); //removes the given node and all its descendants from tree.
8. void displayAncestors(T node); //display ancestors of the given node
9. void displayDescendents(T node); //display descendents of the given node
10. void heightOfTree(); //returns the height (actual height) of tree.
11. void preOrder(); // do the VLR of tree.
12. void postOrder(); // do the LRV of tree.
13. void inOrder(); // do the LVR of tree.
14. void levelOrder(); // do the level order traversal of tree.
15. void displayLevel(int levelNo); // display the nodes on a particular level number.
16. int findLevelOfNode(T node); // returns the level/depth of given node.
17. void displayParenthesizedView(); // display the tree in Parenthesized form.

/*

For Example the parenthesize view of the following binary tree will be

A (B (D (, H) E (I (K ,) J)) C (, F))



*/