|  |  |
| --- | --- |
| **Data Structures & Algorithms**  Diploma in IT, CSF  Year 2 (2024/25) Semester 4 | **Week 13** |
| **0.5 Hour** |
| **Tutorial 10 – AVL Trees** | |



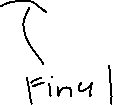
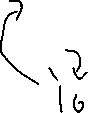
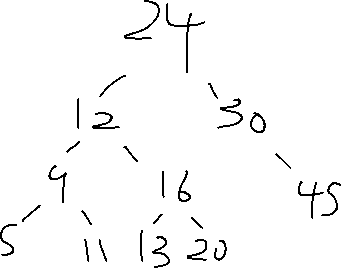
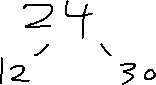
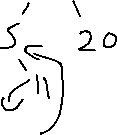
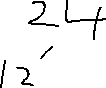
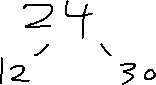
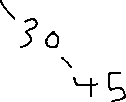
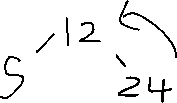
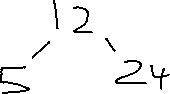
1. An AVL tree is to be created with the items given below.

24, 12, 5, 30, 20, 45, 11, 13, 9, 16

Assuming the items are inserted in the order given (i.e. from left to right),

1. Show the intermediate trees at each stage of rotation.

Explain your workings.



1. Draw the final AVL tree.



1. Write a method that will traverse the final tree in the following manner:

5 11 9 13 20 16 12 45 30 24

You may assume that the node of the tree has been defined as follows:

struct AVLNode

{

ItemType item;

AVLNode\* left; // left subtree

AVLNode\* right; // right subtree

}

void postOrderTraversal(AVLNode\* root)

{

    if (root == nullptr)

        return;          // Base case: empty tree

    // 1. Traverse the left subtree

    postOrderTraversal(root->left);

    // 2. Traverse the right subtree

    postOrderTraversal(root->right);

    // 3. Process the current node (e.g., print its value)

    cout << root->item << " ";

}

1. An AVL tree is to be created with the items given below. Draw the final AVL tree.



|  |  |
| --- | --- |
| INSERT | 7 |
| INSERT | 2 |
| INSERT | 8 |
| INSERT | 6 |
| INSERT | 9 |
| **INSERT** | **3** |
| **INSERT** | **4** |
| **INSERT** | **5** |
| **DELETE** | **9** |

