Analysis and Design Of Algorithms

Name: *Francis Muriithi*

Reg no: *EB3/67225/23*

TYPE OF SORT

1. Tree sort

EXPLANATION

What Is Tree Sort?

- Tree Sort is a sorting algorithm based on the concept of a Binary Search Tree (BST).
- Main Idea :
  - Insert all elements into a Binary Search Tree.
  - Perform an in order traversal of the tree.
  - The traversal gives the elements in sorted order.

PROGRAM FUNCTION

```
#include <iostream>

using namespace std;

// Define structure for Binary Search Tree node

struct Node {

    int data;

    Node* left;

    Node* right;


    Node(int value) {

        data = value;

        left = nullptr;

        right = nullptr;

    }

};
```

```
// Insert a value into Binary Search Tree
Node* insert(Node* root, int value) {
    if (root == nullptr) {
        return new Node(value);
    }

    if (value < root->data) {
        root->left = insert(root->left, value);
    } else {
        root->right = insert(root->right, value);
    }

    return root;
}

// Inorder traversal to store sorted elements
void inorder(Node* root, int arr[], int &index) {
    if (root == nullptr)
        return;

    inorder(root->left, arr, index);
    arr[index++] = root->data;
    inorder(root->right, arr, index);
}
```

```
// Tree Sort function
void treeSort(int arr[], int n) {
    Node* root = nullptr;

    // Step 1: Insert all elements into Binary Search Tree
    for (int i = 0; i < n; i++) {
        root = insert(root, arr[i]);
    }

    // Step 2: Inorder traversal to get sorted array
    int index = 0;
    inorder(root, arr, index);
}

// Main function
int main() {
    int arr[] = {7, 3, 9, 1, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Original array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    treeSort(arr, n);

    cout << "\nSorted array: ";
    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";


    return 0;

}
```