

ASSIGNMENT NO – 08

Given sequence $k = k_1 < k_2 < \dots < k_n$ of n sorted keys, with a search probability p_i for each key k_i . Build the Binary search tree that has the least search cost given the access probability for each key

```
#include<iostream>
using namespace std;

class node{
    int data;
    node *left, *right;
    friend class obst;

    node(int x){
        data = x;
        left= NULL;
        right = NULL;
    }
};

class obst{
    node *root;

public:
    obst(){
        root = NULL;
    }

    double w[10][10]; //weight
    double c[10][10]; //cost
    double r[10][10]; //root
    double k;

    node *create_tree(int i, int j, node* root){
        node *t = NULL;
        if(i != j){
            k = r[i][j]; //root assigning as k
            t = new node(k); //new node creation
            t->left= create_tree(i, k-1, root); //calculate left subtree
            t->right = create_tree(k, j, root); //calculate right subtree
            return t;
        }
        else{
            return NULL;
        }
    }
};
```

```

}

void DisplayTable(){
    for(int i = 0;i<=3;i++){
        cout<<"W "<<i<<","<<i<<=" "<<w[i][i]<<" | "<<"C "<<i<<","<<i<<="
"<<c[i][i]<<" | "<<"R "<<i<<","<<i<<=" "<<r[i][i]<<endl;
    }

    cout<<endl<<endl;

    for(int m =1;m<=3;m++){//Calculate the cost of the tree for more than 1
Nodes.
        for(int i = 0; i<=3-m;i++){
            int j = i + m;
            cout<<"W "<<i<<","<<j<<=" "<<w[i][j]<<" | "<<"C "<<i<<","<<j<<="
"<<c[i][j]<<" | "<<"R "<<i<<","<<j<<=" "<<r[i][j]<<endl;
        }
    }

}

void inorder(node *root){ //inorder traversal
    if(root == NULL){
        return;
    }

    inorder(root->left);
    cout<<root->data<<" ";
    inorder(root->right);
}

void inorder()
{
    this->inorder(root);
    cout<<endl;
}

void calculate_wt(double p[], double q[] ,int n){
    int k;
    // p is array of Probability of Successful search
    // q is array of unsuccessfull search
    // n is number of identifier
    for(int i = 0;i < n;i++){
        w[i][i] = q[i]; //calculates the cost for
        r[i][i] = 0; // NULL Tree & Tree with
        c[i][i] = 0; // One node
    }
}

```

```

        w[i][i+1] = q[i] + q[i+1] + p[i+1];
        r[i][i+1] = i+1;
        c[i][i+1] = w[i][i+1];
    }

    w[n][n] = q[n];
    r[n][n] = 0;
    c[n][n] = 0;

    for(int m = 2; m <= n; m++)//calculate the cost for Trees with More than 1
Node
    for(int i = 0; i <= n - m; i++){
        int min = 999;
        int j = i + m;
        w[i][j] = w[i][j-1] + p[j] + q[j];
        for(int k1 = i+1; k1<=j; k1++){
            int sum1= c[i][k1-1] + c[k1][j];
            if(sum1<min){
                min = sum1;
                k = k1;
            }
        }
        c[i][j] = w[i][j] + c[i][k-1] + c[k][j];
        r[i][j] = k;
    }

    root = create_tree(0, n, root);
}

};

int main(){
    obst tree;
    int ch;
    double p[4] = {0, 0.4 ,0.1, 0.2};
    double q[4] = {0.1,0.04,0.09, 0.07};
    do{
        cout<<"\n*****MENU*****"<<endl;
        cout<<"1.Calculate Weight"<<endl;
        cout<<"2.Display Table"<<endl;
        cout<<"3.Inorder Traversal"<<endl;
        cout<<"4.EXIT";
        cout<<"\nEnter your Choice:"<<endl;
    }

```

```

        cin>>ch;
        switch(ch)
        {
        case 1:
            tree.calculate_wt(p,q,3);
            cout<<"Tree is Calculated";
            break;
        case 2:cout<<"*****Displaying the Table*****"<<endl;;
            tree.DisplayTable();;
            break;
        case 3:cout<<"Inorder Traversal is:";
            tree.inorder();
            break;
        case 4:cout<<"\nThank you for using the Program";
            break;

        }
    }while(ch!=4);
//;

    return 0;
}

```

```

C:\Users\user\Downloads\21474_C08T.exe
*****MENU*****
1.Calculate Weight
2.Display Table
3.Inorder Traversal
4.EXIT
Enter your Choice:
1
Tree is Calculated
*****MENU*****
1.Calculate Weight
2.Display Table
3.Inorder Traversal
4.EXIT
Enter your Choice:
2
*****Displaying the Table*****
W 0,0= 0.1 | C 0,0= 0 | R 0,0= 0
W 1,1= 0.04 | C 1,1= 0 | R 1,1= 0
W 2,2= 0.09 | C 2,2= 0 | R 2,2= 0
W 3,3= 0.07 | C 3,3= 0 | R 3,3= 0

W 0,1= 0.54 | C 0,1= 0.54 | R 0,1= 1
W 1,2= 0.23 | C 1,2= 0.23 | R 1,2= 2
W 2,3= 0.36 | C 2,3= 0.36 | R 2,3= 3
W 0,2= 0.73 | C 0,2= 0.96 | R 0,2= 1
W 1,3= 0.5 | C 1,3= 0.86 | R 1,3= 2
W 0,3= 1 | C 0,3= 1.86 | R 0,3= 1

*****MENU*****
1.Calculate Weight
2.Display Table
3.Inorder Traversal
4.EXIT
Enter your Choice:
3
Inorder Traversal is: 1 2 3

*****MENU*****
1.Calculate Weight
2.Display Table
3.Inorder Traversal
4.EXIT
Enter your Choice:

```