

SC165

Problem Statement:

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>
#include<stdio.h>
#include<bits/stdc++.h>
using namespace std;

#define ROW 10
#define COL 10

class OBST{
    int table[ROW][COL],keys[ROW],freq[ROW],nk,Total_wt,root;
public:
    OBST(){
        keys[0]=freq[0]=0;
    }
    void acceptkeys_freq();
    void createOBST();
    void minCost();
    int sum(int i,int j);
    void display();
};

void OBST::acceptkeys_freq(){
    cout<<"\n Enter no. of keys:";
    cin>>nk;
    cout<<"\n enter keys and their frequencies:";

    for(int i=1;i<=nk;i++){
        cout<<"\nEnter key:";
        cin>>keys[i];
        cout<<"\nEnter frequency:";
        cin>>freq[i];

    }
}

void OBST::display(){
    for(int i=0;i<=nk;i++){
        for(int j=0;j<=nk;j++){
            if(i==j)
                cout<<"\n";
            else
                cout<<"\t";
        }
    }
}
```

```

cout<<table[i][j]<<" ";
}

cout<<"\n";
}

}

int OBST::sum(int i,int j)
{
    int s=0;
    for(int k=i;k<=j;k++)
        s+=freq[k];
    return s;
}

void OBST::minCost(){
    int j,r,c;
    for(int L=2;L<=nk;L++){
        for(int i=0;i<=nk-L;i++){
            j=i+L;
            int min=INT_MAX;
            Total_wt=sum(i,j);
            for(r=i;r<=j;r++){
                c=table[i][r-1]+table[r][j]+Total_wt;
                cout<<"\n Cost:"<<c<<"\n";
                if(c<min)
                    min=c;
            }
            table[i][j]=min;
            //display()
        }
    }
}

void OBST::createOBST(){
    int i,j,r;
    //case:1 j-i=0
    for(i=0;i<nk;i++){
        table[i][i]=0;
    }
    //case:2 j-i=1
    j=1;
    for(i=0;i<nk;i++){
        table[i][j]=freq[j];
        j++;
    }
}

```

```
        }
    //case3:j-1=2
    minCost();
}
int main(){
    OBST bt;
    cout<<"\n OBST:\n";
    bt.acceptkeys_freq();
    bt.createOBST();
    bt.display();
}
```

Output

```
/tmp/G1A20Aqwuk.o

OBST:

Enter no. of keys:4

enter keys and their frequencies:
Enter key:10

Enter frequency:4

Enter key:20

Enter frequency:2

Enter key:30

Enter frequency:6

Enter key:40

Enter frequency:3

Cost:673287806

Cost:8

Cost:10

Cost:12

Cost:30878

Cost:30874

Cost:11

Cost:14
```

```
Cost:14  
  
Cost:17  
  
Cost:30872  
  
Cost:24  
  
Cost:22  
  
Cost:30880  
  
Cost:21  
  
Cost:30886  
  
Cost:20  
  
Cost:27  
  
Cost:673337295  
  
Cost:35  
  
Cost:30  
  
Cost:26  
  
Cost:37  
0 4 8 22 26  
0 30860 2 12 20  
0 0 0 6 11  
0 30860 673337472 30860 3  
6 16 673335808 30860 0
```