Problem Statement:

Given sequence k = k1 <k2 < … <kn of n sorted keys, with a search probability pi for each key ki . Build the Binary search tree that has the least search cost given the access probability for each key?

#include<iostream>

using namespace std;

int find(int, int);

void print(int, int);

int c[20][20], w[20][20], r[20][20], p[20], q[20], k, m, i, j, n;

char idtr[10][7];

int main()

{

cout<<"\nEnter number of identifiers: ";

cin>>n;

for(i = 1; i <= n; i++)

{

cout<<"Enter Identifier "<<i<<": ";

cin>>idtr[i];

}

for(i = 1; i <= n; i++)

{

cout<<"Enter successful probability of "<<i<<": ";

cin>>p[i];

}

for(i = 0; i <= n; i++)

{

cout<<"Enter unsuccessful probability of "<<i<<": ";

cin>>q[i];

}

for(i = 0; i <= n; i++)

{

w[i][i] = q[i];

c[i][i] = r[i][i] = 0;

cout<<"\nW: "<<w[i][i]<<" | c: "<<c[i][i]<<" | r: "<<r[i][i];

}

for(i = 0; i < n; i++)

{

j = i + 1;

w[i][j] = p[j] + q[j] + q[i];

c[i][j] = w[i][j];

r[i][j] = j;

cout<<"\nW: "<<w[i][j]<<" | c: "<<c[i][j]<<" | r: "<<r[i][j];

}

for(m = 2; m <= n; m++)

{

for(i = 0; i <= n-m; i++)

{

j = i + m;

w[i][j] = p[j] + q[j] + w[i][j-1];

c[i][j] = 1000;

for(k = i + 1; k <= j; k++)

{

int cost = c[i][k-1] + c[k][j];

if(cost < c[i][j])

{

c[i][j] = cost;

r[i][j] = k;

}

}

c[i][j] += w[i][j];

cout<<"\nW: "<<w[i][j]<<" | c: "<<c[i][j]<<" | r: "<<r[i][j];

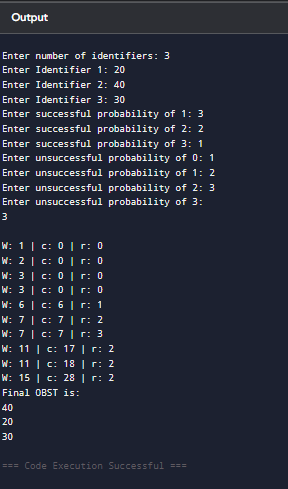
}

}

cout<<"\nFinal OBST is: ";

print(0, n);

return 0;

}

void print(int i, int j)

{

if(i < j)

cout<<"\n"<<idtr[r[i][j]];

else

return;

print(i, r[i][j] - 1);

print(r[i][j], j);

}