**Lab 12**

**Prasanna Natarajan**

**1410110298**

Code:

/\*

Name        :   Prasanna Natarajan

Roll Number :   1410110298

Inputs      :   A vector p and q of length N and N+1

Outputs     :   The matrices containing the root and e value of the most optimal BST

Description :   This algorithm is taken from Introduction to Algorithm By Cormen Pg.402

\*/

#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

#define N 5

// defining global variables

double e[N+2][N+1];

double w[N+2][N+1];

int root[N+1][N+1];

//function declarations

void optimal\_bst(double p[],double q[], int n);

int main(){

   // user inputs

   double TempP[N] = {0.15,0.10,0.05,0.1,0.2};

   double q[N+1] = {0.05,0.10,0.05,0.05,0.05,0.10};

   int i;

   double p[N+1];

   p[0] = 0;

   for(i=0;i<N;i++){

       p[i+1] = TempP[i];

       //printf("%lf",p[i+1]);

   }

   //initialising all the matrices

   int j;

   for(i=0;i<=N+1;i++){

       for(j=0;j<N+1;j++){

           e[i][j] = 0;

       }

   }

   for(i=0;i<N+1;i++){

       for(j=0;j<N+1;j++){

           root[i][j] = 0;

       }

   }

   for(i=0;i<N+2;i++){

       for(j=0;j<N+1;j++){

           w[i][j] = 0;

       }

   }

   // calling optimal bst to populate the matrices

   optimal\_bst(p,q,N);

   printf("The matrix e is: \n");

   for(i=1;i<=N+1;i++){

       for(j=0;j<N+1;j++){

           printf("%lf ",e[i][j]);

       }

       printf("\n");

   }

   printf("The matrix root is: \n");

   for(i=1;i<N+1;i++){

       for(j=1;j<N+1;j++){

           printf("%d ",root[i][j]);

       }

       printf("\n");

   }

   printf("The matrix w is: \n");

   for(i=0;i<N+2;i++){

       for(j=0;j<N+1;j++){

           printf("%lf ",w[i][j]);

       }

       printf("\n");

   }

   return 0;

}

void optimal\_bst(double p[],double q[], int n){

   int i =0;

   // initialising some values(diagonal elements) of e and w

   for(i=1;i<n+2;i++){

       e[i][i-1] = q[i-1];

       w[i][i-1] = q[i-1];

   }

   int l;

   for(l=1;l<n+1;l++){

       for(i=1;i<=n-l+1;i++){

           int j = i+l-1;

           e[i][j] = 1000000.0;

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

           int r;

           for(r=i;r<=j;r++){

               double t = e[i][r-1] + e[r+1][j] + w[i][j];

               if(t<e[i][j]){

                   e[i][j] = t;

                   root[i][j] = r;

               }

           }

       }

   }

}

