**OPTIMAL BINARY SEARCH**

#include<conio.h>

#define MAX 10

int main()

{

char ele[MAX][MAX];

int w[MAX][MAX], c[MAX][MAX], r[MAX][MAX], p[MAX], q[MAX];

int temp=0, root, min, min1, n;

int i,j,k,b;

printf("Enter the number of elements:");

scanf("%d",&n);

printf("\n");

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

{

printf("Enter the Element of %d:",i);

scanf("%d",&p[i]);

}

printf("\n");

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

{

printf("Enter the Probability of %d:",i);

scanf("%d",&q[i]);

}

printf("W\t\tC\t\tR\n");

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

{

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

{

if(i == j)

{

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

c[i][j] = 0;

r[i][j] = 0;

printf("W[%d][%d]: %d\tC[%d][%d]: %d\tR[%d][%d]: %d\n",i,j,w[i][j],i,j,c[i][j],i,j,r[i][j]);

}

}

}

printf("\n");

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

{

for(i=0,j=b+1; j < n+1 && i < n+1; j++,i++)

{

if(i!=j && i < j)

{

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

min = 30000;

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

{

min1 = c[i][k-1] + c[k][j] + w[i][j];

if(min > min1)

{

min = min1;

temp = k;

}

}

c[i][j] = min;

r[i][j] = temp;

}

printf("W[%d][%d]: %d\tC[%d][%d]: %d\tR[%d][%d]: %d\n",i,j,w[i][j],i,j,c[i][j],i,j,r[i][j]);

}

printf("\n");

}

printf("Minimum cost = %d\n",c[0][n]);

root = r[0][n];

printf("Root = %d \n",root);

getch();

}

