**Travelling salesman problem**

#include<stdio.h>

int array[20][20],visited[20],n,cost=0;

int main(){

int i,j;

printf("enter the number of nodes..");

scanf("%d",&n);

printf("enter the cost matrix:\n");

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

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

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

visited[i] = 0;

}

mincost(0);

printf("\nthe cost: %d",cost);

return 0;

}

void mincost(int m){

int i,node;

visited[m] = 1;

printf("%d->",m+1);

node = least(m);

if(node == 99999){

node = 0;

printf("%d",node+1);

cost +=array[m][node];

return;

}

mincost(node);

}

int least(int c)

{

int i,mx=99999;

int min=99999,kmin;

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

{

if((array[c][i]!=0)&&(visited[i]==0))

if(array[c][i] < min)

{

min=array[i][c]+array[c][i];

kmin=array[c][i];

mx=i;

}

}

if(min!=99999)

cost+=kmin;

return mx;}

**Longest Common Subsequence**

#include<stdio.h>

#include<string.h>

char x[20],y[20], b[20][20];

int c[20][20],i,j,m,n;

int main(){

printf("enter first string..");

gets(x);

printf("enter second string..");

gets(y);

lcs();

return 0;}

void lcs(){

m = strlen(x);

n = strlen(y);

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

c[i][0] = 0;

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

c[0][j] = 0;

for(i=1;i<=m;i++){

for(j=1;j<=n;j++){

if(x[i-1]==y[j-1]) {

c[i][j] = c[i-1][j-1]+1;

b[i][j] = 'c';}

else if(c[i-1][j]>=c[i][j-1]){

c[i][j] = c[i-1][j];

b[i][j] = 'u';

}

else{

c[i][j] = c[i-1][j];

b[i][j] = 'l'; }

}

print(m,n);}

}

print(int i,int j){

if(i==0 || j==0)

return;

else if(b[i][j]=='c')

{

print(i-1,j-1);

printf(" %c",x[i-1]);

}

else if(b[i][j]=='u'){

print(i-1,j);

}

else

print(i,j-1);

}

**Chain Multiplication**

#include<stdio.h>

int D[100] , M[20][20];

int main(){

int size,i,value;

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

scanf("%d",&size);

printf("Enter the dimensions of matrices ");

for(i=0;i<=size;i++){

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

}

value = chain(1,size);

printf("\nNumber of multiplication: %d\n",value);

printf("\nThe Optimal order is:\n");

print\_order(1,size);

return 0;

}

int chain(int i,int j){

if(i==j)

return 0;

int k,count;

int min=1000000;

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

count=chain(i,k) + chain(k+1,j) + D[i-1]\*D[k]\*D[j];

if(count<min){

min=count;

M[i][j]=k;

}

}

return min;

}

void print\_order(int i,int j)

{

if(i==j)

printf("M%d",i);

else

{

printf("(");

print\_order(i,M[i][j]);

print\_order(M[i][j]+1,j);

printf(")");

}

}