

Competitive Programming

(<https://github.com/MalavikaJayakumar/Competitive-Programming-Problems>)

1. Floyd Warshall Algorithm

Code:

```
#include<iostream>

using namespace std;
#define inf 9999
int n;
void printmatrix(int *dmatrix)
{
    cout<<"\nThe Shortest path between vertices in matrix form: \n";
    for(int i = 0;i <n ;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(*((dmatrix+(i*n))+j) == inf)
                cout<<"X" <<"\t";
            else
                cout<<*((dmatrix+(i*n))+j)<<"\t";
        }
        cout<<endl;
    }
}

void FloydWarshall(int *graph)
{
    int dmatrix[n][n],i,j,k;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            dmatrix[i][j]= *((graph+(i*n))+j);
        }
    }

    for(k=0;k<n;k++)
    {
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                if(dmatrix[i][k] + dmatrix[k][j]<dmatrix[i][j])
                {
                    dmatrix[i][j]= dmatrix[i][k]+dmatrix[k][j];
                }
            }
        }
    }
}
```

```

    }
    }
    }
    printmatrix((int *)dmatrix);
}

int main()
{
    int e,i,x,y,w;
    cout<<"Enter number of verices and edges : ";
    cin>>n>>e;
    int graph[n][n];
    for(int j=0;j<n;j++)
    {
        for(int k=0;k<n;k++)
        {
            if(j==k)
                graph[j][k]=0;
            else
                graph[j][k]=inf;
        }
    }
    cout<<"Enter start node,end node and weight : \n";
    for(i=0;i<e;i++)
    {
        cin>>x>>y>>w;
        graph[x][y]=w;
    }
    FloydWarshall((int *)graph);
    return 0;
}

```

Output:

```

C:\WINDOWS\SYSTEM32\cmd.exe
Enter number of verices and edges : 5 7
Enter start node,end node and weight :
0 1 7
0 4 4
0 2 1
1 3 6
2 4 2
2 3 3
4 3 5

The Shortest path between vertices in matrix form:
0      7      1      4      3
X      0      X      6      X
X      X      0      3      2
X      X      X      0      X
X      X      X      5      0

```