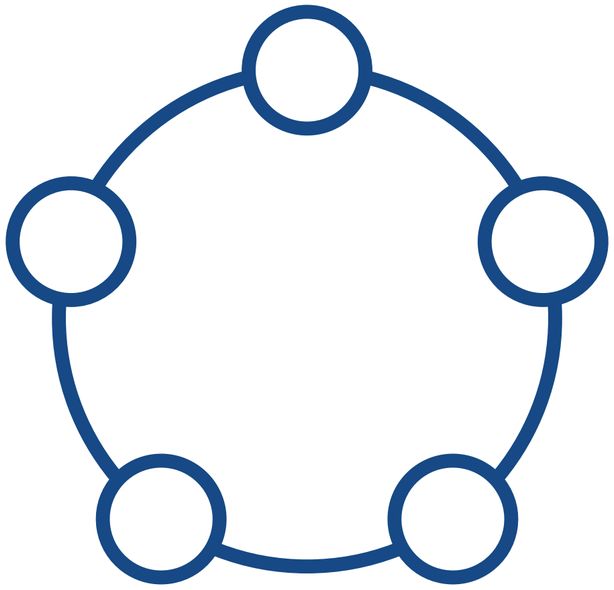

Semana 9

Teoria de Grafos

Ana Luiza Sticca, Gabriel Antônio, Guilherme Cabral e João Pedro Marques

06 All Pairs Shortest Path

All Pairs Shortest Path

Sample Input 1

```
4 3 4
0 1 2
1 2 2
3 3 1
0 2
1 2
3 0
3 3
2 1 2
0 1 100
0 1
1 0
0 0 0
```

Sample Output 1

```
4
2
Impossible
0

100
Impossible
```

All Pairs Shortest Path (versão 1)

```
typedef struct Edge
{

    int u, v, w;

}Ed;

vector< Ed > edges;

int dist[NMAX][NMAX];
int dist2[NMAX][NMAX];

void Solv(int n)
{

    int u, v, w, init, i;

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

        for(i = 0;i < n;i++) dist[init][i] = INF;
        dist[init][init] = 0;
```

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

    for(auto cur : edges)
    {

        u = cur.u;
        v = cur.v;
        w = cur.w;

        if(dist[init][u] != INF && dist[init][v] > dist[init][u] + w)
            dist[init][v] = dist[init][u] + w;

    }

}

for(i = 0;i < n;i++) dist2[init][i] = dist[init][i];
```

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

    for(auto cur : edges)
    {

        u = cur.u;
        v = cur.v;
        w = cur.w;

        if(dist2[init][u] != INF && dist2[init][v] > dist2[init][u] + w)
            dist2[init][v] = dist2[init][u] + w;

    }

}

for(i = 0;i < n;i++)
    if(dist[init][i] != dist2[init][i])
        dist[init][i] = -INF;
```

```cpp
int main()
{

    int n, m, q, u, v, w, i, j, k, l;

    while(cin >> n >> m >> q)
    {

        if(n == 0 && m == 0 && q == 0) break;

        edges.clear();

        while(m--)
        {

            cin >> u >> v >> w;

            edges.push_back({u, v, w});

        }

        Solv(n);

        Solv(n);

        while(q--)
        {

            cin >> u >> v;

            if(dist[u][v] == INF)        cout << "Impossible" << endl;
            else if(dist[u][v] == -INF) cout << "-Infinity" << endl;
            else                         cout << dist[u][v] << endl;

        }

        cout << endl;

}

return 0;
```

All Pairs Shortest Path (versão 2)

```cpp
int main()
{

    int n, m, q, u, v, w, i, j, k;

    while(cin >> n >> m >> q)
    {

        if(n == 0 && m == 0 && q == 0) break;

        for(i = 0;i < n;i++)
            for(j = 0;j < n;j++)
                flody[i][j] = INF;

        for(i = 0;i < n;i++)
            flody[i][i] = 0;

        while(m--)
        {

            cin >> u >> v >> w;

            flody[u][v] = min(flody[u][v], w);
```

```cpp
for(k = 0;k < n;k++)
    for(i = 0;i < n;i++)
        for(j = 0;j < n;j++)
            if(flody[i][k] < INF && flody[k][j] < INF)
                flody[i][j] = min(flody[i][j],
                                  flody[i][k] + flody[k][j]);

for(k = 0;k < n;k++)
    for(i = 0;i < n;i++)
        for(j = 0;j < n;j++)
            if(flody[k][k] < 0 && flody[i][k] != INF && flody[k][j] != INF)
                flody[i][j] = -INF;

while(q--)
{

    cin >> u >> v;

    if(flody[u][v] == INF)          cout << "Impossible" << endl;
    else if(flody[u][v] == -INF)    cout << "-Infinity" << endl;
    else                            cout << flody[u][v] << endl;

}

cout << endl;
```

Exemplos rodados

```
4 3 4
0 1 2
1 2 2
3 3 1
0 2
1 2
3 0
3 3
2 1 2
0 1 100
0 1
1 0
0 0 0
4
2
Impossible
0

100
Impossible
```