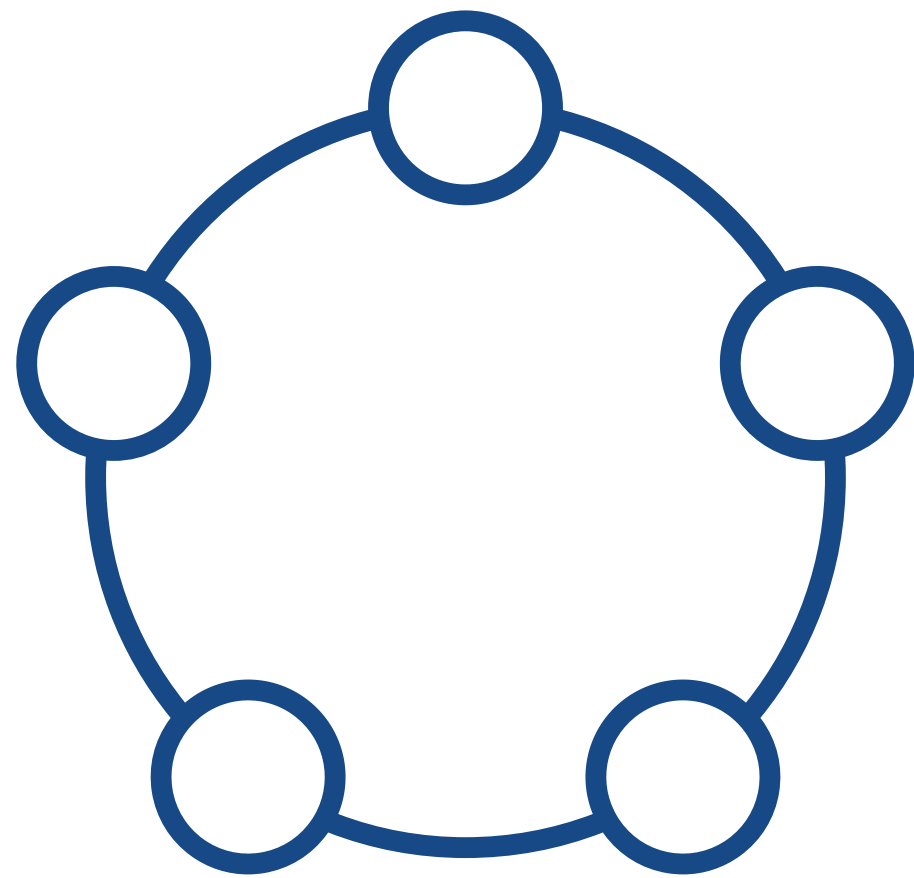


# Semana 8



## Teoria de Grafos



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

# Problema



## 06 Bank Robbery



06

Bank Robbery

# Bank Robbery

**Estrutura de dados:** Listas de Adjacência

**Vértices:** Locais da cidade (incluindo os bancos e as delegacias)

**Arestas:** Estradas que ligam os locais da cidade

# Bank Robbery

## Sample Input

```
5 6 2 1
0 1 5
0 2 2
1 3 6
1 4 1
2 3 4
3 4 3
1 4
2
```

```
5 4 2 1
0 1 5
0 2 2
1 3 6
2 3 4
1 4
2
5 6 2 2
0 1 5
0 2 2
1 3 6
1 4 1
2 3 4
3 4 3
1 4
2 3
```

## Sample Output

```
2 7
1 4
1 *
4
1 4
1
```

```

typedef pair< int, int > pii;
typedef vector< int > vi;
typedef vector< pii > vii;

int n;
vii grafo[NMAX];

int dist[NMAX];

void Dijkstra(vi sources)
{
    int u, v, w, d, i;

    priority_queue< pii, vii, greater< pii > > fila;

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

    for(auto cur : sources)
    {
        dist[cur] = 0;
        fila.push({dist[cur], cur});
    }
}

```

```

while(fila.empty() == false)
{
    d = fila.top().first;
    u = fila.top().second;
    fila.pop();

    if(d != dist[u]) continue;

    for(auto viz : grafo[u])
    {
        w = viz.first;
        v = viz.second;

        if(dist[v] > dist[u] + w)
        {
            dist[v] = dist[u] + w;
            fila.push({dist[v], v});
        }
    }
}

```

```

int main()
{

    int m, b, p, u, v, w, i;

    bool fi;

    vi ba, po, resp;

    while(cin >> n >> m >> b >> p)
    {

        for(i = 0; i < n; i++) grafo[i].clear();
        ba.clear();
        po.clear();
        resp.clear();

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

            cin >> u >> v >> w;

            grafo[u].push_back({w, v});
            grafo[v].push_back({w, u});

        }
    }
}

```

```

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

    cin >> u;

    ba.push_back(u);

}

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

    cin >> u;

    po.push_back(u);

}

Dijkstra(po);

w = -1;

```

```
for(auto cur : ba)
{

    if(w < dist[cur])
    {

        w = dist[cur];
        resp.clear();

    }

    if(w == dist[cur]) resp.push_back(cur);

}

sort(resp.begin(), resp.end());
```

```
cout << resp.size() << " ";

if(w == INF)    cout << "*" << endl;
else           cout << w << endl;

fi = true;

for(auto cur : resp)
{

    if(fi == false) cout << " ";
    fi = false;

    cout << cur;

}

cout << endl;
```



# Exemplos rodados

```
5 6 2 1
0 1 5
0 2 2
1 3 6
1 4 1
2 3 4
3 4 3
1 4
2
5 4 2 1
0 1 5
0 2 2
1 3 6
2 3 4
1 4
2
5 6 2 2
0 1 5
0 2 2
1 3 6
1 4 1
2 3 4
3 4 3
1 4
2 3 2 7
1 4
1 *
4
```