

Dijkstra

Algorytmika od podstaw

20 listopada 2018

```
#include<iostream>
#include<vector>
#include<set>
using namespace std;

const int N = 100 * 1000 + 15;

int n,m;
int odl[N];
vector<int> G[N];
vector<int> war[N];
int nies = 1000000000;

struct porset{
    bool operator()(int a, int b){
        if(odl[a]==odl[b]){
            return a<b;
        }
        return odl[a]<odl[b];
    }
};

set<int, porset>s;

void dijkstra(int x)
{
    for(int i=1;i<=n;i++){
        odl[i]=nies;
    }
    odl[x]=0;
    s.insert(x);

    while(!s.empty())
    {
        x=*s.begin();
        s.erase(x);
        for(int i=0;i<G[x].size();i++)
        {
            int a=G[x][i];
            if(odl[x]+war[x][i]<odl[a])
            {
                if(odl[a]<nies)
                    s.erase(a);
            }
        }
    }
}
```

```

        odl[a]=odl[x]+war[x][i];
        s.insert(a);
    }
}

int main()
{
    ios_base::sync_with_stdio(0);
    int x;
    cin >> n >> m >> x;
    for(int i=0;i<m;i++)
    {
        int a,b,c;
        cin >> a >> b >> c;
        G[a].push_back(b);
        war[a].push_back(c);
        G[b].push_back(a);
        war[b].push_back(c);
    }

    dijkstra(x);

    cout << odl[n] << '\n';
    return 0;
}

```