

```

1 #include "bits/stdc++.h"
2
3 using namespace std;
4
5 const int N = 1000000 + 15;
6 const int INF = 1000000000 + 15;
7 vector<int> G[N], wart[N];
8 int n, m, odl[N];
9 bool odw[N];
10
11 /*Prosty kod
12   Który wczytuje graf z krawędziami ważonymi
13   Puszczą na nim dijkstre
14   i zaznacza odległość do każdego)*/
15
16 void Dijkstra(int x) {
17     // ustawienie wszystkich odległości na nieskończoność, czasami się nie
18     // przydaje
19     for (int i = 0; i < N; i++)
20         odl[i] = INF;
21
22     odl[x] = 0;
23     set< pair <int, int> > s;
24     s.insert(make_pair(0, x));
25
26     while (s.size() != 0) {
27         int x = s.begin()->second;
28         int akt_odl = s.begin()->first;
29         // normalnie trzeba by było pisać
30         // x = *s.begin().first
31         // ale ' -> ' zastępuje '* ___ .'
32         s.erase(s.begin());
33
34         if (odw[x] == true)
35             continue;
36         odw[x] = true;
37
38         for (int i = 0; i < G[x].size(); i++) {
39             int pom_odl = akt_odl + wart[x][i];
40             int v = G[x][i];
41
42             if (pom_odl < odl[v]) {
43                 s.insert(make_pair(pom_odl, v));
44                 odl[v] = pom_odl;
45             }
46         }
47     }
48 }
49
50 int main() {
51     cin >> n >> m;
52
53     for (int i = 0; i < m; i++) {
54         int a, b, c;
55         cin >> a >> b >> c;
56         G[a].push_back(b);
57         G[b].push_back(a);
58         wart[a].push_back(c);
59         wart[b].push_back(c);
60     }
61
62     Dijkstra(1);
63
64     for (int i = 1; i <= n; i++)
65         cout << odl[i] << "\n";
66 }

```