**Дейкстра**

for (int i = 1; i <= n; i++)

{

if (i == st) {way.insert(make\_pair(0, i));}

else {way.insert(make\_pair(INF, i));w[i]=INF;}

}

w[st] = 0;

dijkstra(st);

const int INF = 1000000000, MAX\_SIZE = 100000+5;

int n, m, st, fn;

int w[MAX\_SIZE], used[MAX\_SIZE];

typedef pair <int, int> pa;

set <pa> way;

set <pa>:: iterator iter, it;

map <int,vector <pa> > ed;

void dijkstra(int v)

{

used[v] = 1;

way.erase(way.begin());

for (int i = 0; i < ed[v].size(); i++)

{

pa to = ed[v][i];

int tmp = w[v]+to.first;

if (used[to.second]==0)

{

//used[to.second]==1;

if (w[to.second] > tmp)

{

way.erase(make\_pair(w[to.second], to.second));

way.insert(make\_pair(tmp, to.second));

w[to.second] = tmp;

}

}

}

if (way.size() > 0){iter = way.begin();dijkstra((\*iter).second);}

}