**Форд-Беллман**Пусть дан ориентированный взвешенный граф с вершинами и рёбрами, и указана некоторая вершина. Требуется найти длины кратчайших путей от вершины до всех остальных вершин.

structedge { int a, b, cost; };

int n, m, v;

vector <edge> e;

const intINF = 1000000000;

void solve() {

vector<int> d (n, INF);

d[v]= 0;

for (inti=0; i<n-1; ++i)

for (intj=0; j<m; ++j)

if (d[e[j].a]< INF)

d[e[j].b]= min (d[e[j].b], d[e[j].a]+ e[j].cost);

}

**Алгоритм Левита нахождения кратчайших путей от заданной вершины до всех**

**остальных вершин**

typedef pair<int,int> rib;

typedef vector < vector<rib> > graph;

const int inf = 1000\*1000\*1000;

int main()

{

int n, v1, v2;

graph g (n);

... чтение графа...

vector<int> d (n, inf);

d[v1] = 0;

vector<int> id (n);

deque<int> q;

q.push\_back (v1);

vector<int> p (n, -1);

while (!q.empty())

{

int v = q.front(), q.pop\_front();

id[v] = 1;

for (size\_t i=0; i<g[v].size(); ++i)

{

int to = g[v][i].first, len = g[v][i].second;

if (d[to] > d[v] + len)

{

d[to] = d[v] + len;

if (id[to] == 0)

q.push\_back (to);

else if (id[to] == 1)

q.push\_front (to);

p[to] = v;

id[to] = 1;

}

}

}

... вывод результата...

}