首先输入数据 第一号：2 3 2（顶点2到顶点3距离为2，跟dijkstra的差不多）

第二号：1 2 -3

第三号：1 5 5

第四号：4 5 2

第五号：3 4 3

首先上代码，核心代码只有4行，很简洁。

for(k = 1;k <= n - 1;k++)

{

for(i = 1;i <= m;i++)

{

if(dis[v[i]] > dis[u[i]] + w[i])

{

dis[v[i]] = dis[u[i]] + w[i];

}

}

}

if语句很眼熟对不对，要更新的就是dis[v[i]]（源顶点1到v[i]点的距离），判断dis[u[i]] + w[i]（u[i]到v[i]的距离）是否比dis[v[i]]小，小的话就更新，这种操作就叫松弛。

那外面两个循环是什么回事呢？

第二个循环就是遍历每一个边，很好理解

第一个循环就是根据顶点数进行循环，第一轮循环结束后，得到的是从1号顶点出发，可以直接到达顶点的最短路径。第二轮循环结束后，得到的是从1号顶点出发，最多经过一个顶点的最短路径。以此类推，进行n-1轮。

到最后 ，这个版本的bellman-ford算法有很多地方要优化的地方。

最短路径最多有n-1条边，在每一轮，就是上面的第一个循环，我们都要执行松弛操作，有些边求得了最短路径了，在接下来的循环不会变，但我们还会处理它，这里就浪费了时间，可以优化。所以就有了bellman-ford的队列优化。