**Bellman-Ford算法**

Bellman-Ford算法能在更普遍的情况下（存在负权边）解决单源点最短路径问题。对于给定的带权（有向或无向）图 G=（V,E），其源点为s，加权函数 w 是边集 E 的映射。对图G运行Bellman-Ford算法的结果是一个布尔值，表明图中是否存在着一个从源点s可达的负权回路。若不存在这样的回路，算法将给出从源点s到图G的任意顶点v的最短路径d[v]。

Bellman-Ford算法流程分为三个阶段：

（1）初始化：将除源点外的所有顶点的最短距离估计值 d[v] ←+∞, d[s] ←0;  
（2）迭代求解：反复对边集E中的每条边进行松弛操作，使得顶点集V中的每个顶点v的最短距离估计值逐步逼近其最短距离；（运行|v|-1次）  
（3）检验负权回路：判断边集E中的每一条边的两个端点是否收敛。如果存在未收敛的顶点，则算法返回false，表明问题无解；否则算法返回true，并且从源点可达的顶点v的最短距离保存在 d[v]中。

算法描述如下：

Bellman-Ford(G,w,s) ：boolean   //图G ，边集 函数 w ，s为源点

for each vertex v ∈ V（G） do        //初始化 1阶段

d[v] ←+∞

d[s] ←0;                            //1阶段结束

for i=1 to |v|-1 do                  //2阶段开始，双重循环。

for each edge（u,v） ∈E(G) do    //边集数组要用到，穷举每条边。

If d[v]> d[u]+ w(u,v) then     //松弛判断

d[v]=d[u]+w(u,v)            //松弛操作   2阶段结束

for each edge（u,v） ∈E(G) do

If d[v]> d[u]+ w(u,v) then

Exit false

Exit true

适用条件和范围：

1. 单源最短路径(从源点s到其它所有顶点v);
2. 有向图&无向图(无向图可以看作(u,v),(v,u)同属于边集E的有向图);
3. 边权可正可负(如有负权回路输出错误提示);
4. 差分约束系统;