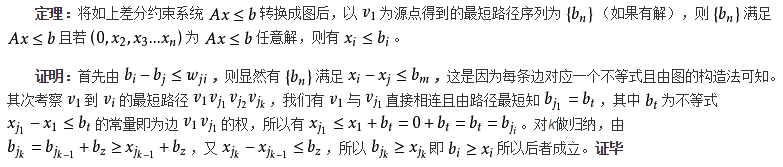
**差分约束系统：**

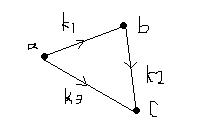
**如果一个系统由n个变量和m个约束条件组成，形成m个形如ai-aj≤k的不等式(i,j∈[1,n],k为常数),则称其为差分约束系统(system of difference constraints)。亦即，差分约束系统是求解关于一组变量的特殊不等式组的方法。如果一个系统由n个变量和m个约束条件组成，形成m个形如ai-aj≤k的不等式(i,j∈[1,n],k为常数),则称其为差分约束系统(system of difference constraints)。亦即，差分约束系统是求解关于一组变量的特殊不等式组的方法。**

**求解差分约束系统，可以转化成图论的[单源最短路径](http://baike.baidu.com/view/1939816.htm" \t "http://baike.baidu.com/_blank)（或最长路径）问题。**

**观察xj-xi<=bk，会发现它类似最短路中的三角不等式d[v]<=d[u]+w[u,v]，即d[v]-d[u]<=w[u,v]。因此，以每个变量xi为结点，对于约束条件xj-xi<=bk，连接一条边(i,j)，边权为bk。我们再增加一个源点s,s与所有定点相连，边权均为0。对这个图，以s为源点运行Bellman-ford算法（或SPFA算法），最终{d[ i]}即为一组可行解。**



**比如给出三个不等式,b-a<=k1,c-b<=k2,c-a<=k3,求出c-a的最大值,我们可以把a,b,c转换成三个点，k1，k2，k3是边上的权，如图**

****

**由题我们可以得知，这个有向图中，由题b-a<=k1,c-b<=k2,得出c-a<=k1+k2,因此比较k1+k2和k3的大小，求出最小的就是c-a的最大值了**

**根据以上的解法，我们可能会猜到求解过程实际就是求从a到c的最短路径，没错的....简单的说就是从a到c沿着某条路径后把所有权值和k求出就是c -a<=k的一个**

**推广的不等式约束，既然这样，满足题目的肯定是最小的k，也就是从a到c最短距离...**

**Points:**

**1. 如果要求最大值想办法把每个不等式变为标准x-y<=k的形式,然后建立一条从y到x权值为k的边,变得时候注意x-y<k =>x-y<=k-1。如果要求最小值的话,变为x-y>=k的标准形式，然后建立一条从y到x的k边，求出最长路径即可。**

**2.如果权值为正，用dj，spfa，bellman都可以，如果为负不能用dj，并且需要判断是否有负环，有的话就不存在。**