差分约束系统转变为最短路的关键是其对原来最短路中的距离数组 Dist 含义的定义,以及依据此定义构筑的边。一般而言,Dist 数组是一个延伸意义上的数组,它不直接对应题目数据的含义。由差分约束系统的定义,Dist[k]往往是前 k 个数的和,这样可以用 Dist[x]-Dist[k-1]表示 sigma(k,k+1,.....,x),对应地 Dist[x]-Dist[k-1]<=Weight[i]。最短路中的三角形法则判断转移方程是形如

Dist[new]>Dist[now]+Edge[now,new]的,所以对于差分约束系统的限制条件,应该通过变形转变为类似的形式。如:限制条件转化为形如Dist[Bi]<=Dist[Ai]+K,则构图时应该转化为从 Ai 向 Bi 引一条单向边,边权为 K; 限制条件转化为形如 Dist[Ci]<=Dist[Di]-K,则构图时应该转化为从 Di 向 Ci 引一条单向边,边权为-K。

求最大值,做最短路径:一开始 Dist 数组被赋初始值为极大,随后每一次更新它的值的时候都是依据不等式进行的约束(从而数组值回到正常范围并且逐渐达到最优解),而最短路径确保其被约束所减小的值最小,所以保证最后得到最大的值。

求最小值,做最长路径:一开始 Dist 数组被赋初始值为 0,随后每一次更新它的值都是依照不等式进行的约束,而 Dist 数组的负值体现了其被约束条件累计减少的数值,而最长路确保其能得到最多的约束而得到最小值。但是注意,在 SPFA 中的三角比较依然不需要改变,还是 if Dist[new]>Dist[now]+Edge[now,new] then Dist[new]:

=Dist[now]+Edge[now,new],因为约束条件在 Edge 数组里的体现是负权,所以会越更新越小,而不需要因为最长路而更换符号。

求存在性,则判断负环:如果一组约束最后使得某个点自身对自身有约束,也就是形如 a[i]>(<)a[i]+k,明显在数学上不成立,故无满足约束的解存在。而负环则意味着在这条约束路径上对某一个点可以产生一个自身对自身的约束(因为存在值被更新了一定是由约束所规定的),则显然不成立。

POJ1201【基础】

题目大意:

给定一个整数区间[Ai,Bi], Ai<Bi, 要求在此区间中至少有 Ci 个点被选择。如今有 N (1<=N<=50000) 个区间, 求满足这 N 个区间的点集中最少要有多少个点。

输入:

第一行一个正整数 N。

接下来 N 行每一行有两个用空格分隔的整数 Ai、Bi(0<=ai, bi<=50000)和 Ci (1<=ci<=bi-ai+1)。

输出:

一行,表示点集中需要有的最小的点的数量。

题解:

如果用 Dist[i]表示[0,i-1]最少要有多少个整数点,由题意得 Dist[b+1]-Dist[a]>=Ci,即 Dist[a]-Dist[b+1]<=-Ci。自然情况的约束条件为 Dist[i+1]-Dist[i]<=1 和 Dist[i+1]-Dist[i]>=0。结果是 Dist[n]-Dist[0],n 为最大点的编号。

POJ1716【基础】

题目大意:

给定一个整数区间[Ai,Bi], Ai<Bi, 要求在此区间中至少有两个点被选择。如今有 N (1<=N<=10000) 个区间, 求满足这 N 个区间的点集中最少要有多少个点。

输入:

第一行一个整数 N。

接下来 N 行每一行有两个用空格分隔的整数 Ai 和 Bi (0<=ai, bi<=10000)。

输出:

一行,表示点集中需要有的最小的点的数量。

题解:

POJ1201 的特殊版。

POJ3159【基础】

见《【重拾算法系列】总序&一、POJ 最短路系列题目题解》中 POJ3159 题的 题解。

POJ3169【中等】

题目大意:

农夫 John 有 N(2<=N<=1000)只奶牛依次编号为 1 到 N,现在它们站在一条直线上等待被饲喂。它们排列的顺序是按编号递增的,即小的在前大的在后。又由于它们很(抱歉我不知道怎么翻译 pushy 对应上下文),所以它们可以有几头牛打横并排站在同一排位,也允许两头牛之间隔着一个或以上的排位。前后两个排位之间我们定义为一个单位距离。奶牛们在进食的时候也喜欢和自己的奶牛呆在一起而远离自己讨厌的奶牛。有些奶牛想和别的某只奶牛在一定的距离内,另外一些想和某只奶牛在一定的距离外。一个 ML 列表(1<=ML<=10000)指出有哪些奶牛之间是互相喜欢的以及它们之间所允许的最大距离;另一个列表 MD(1<=MD<=10000)指出有哪些奶牛之间是互相讨厌的以及他们之间所允许的最小间隔。

现在你需要去计算,如果可以满足所有奶牛的需求,那么第一头牛和最后一头牛之间可能的最大距离。

输入:

第一行是三个空格分隔的整数 N、ML 和 MD。

接下来 ML 行,每一行包含三个空格分隔的正整数 A、B(1<=A<B<=N)和 D(1<=D<=1000000),表示编号为 A 和 B 的两头奶牛之间允许的最大距离。接下来 MD 行,每一行包含三个空格分隔的正整数 A、B(1<=A<B<=N)和 D(1<=D<=1000000),表示编号为 A 和 B 的两头奶牛之间允许的最小间隔。

输出:

一行,一个整数:如果没有可以满足的解决方案,输出-1;如果编号为 1 和编号为 N 的牛之间的距离可以是任意远,输出-2;剩余情况下输出编号为 1 和编号为 N 的牛之间可能的最大距离。

题解:

关于这题的思考与题解,我强烈推荐这个: http://www.2cto.com/kf/201207/143735.html, 因为此文不仅是对本题的分析,还可以帮助我们更好地了解差分约束系统。因此我不再赘述。

POJ1364【中等】

题目大意:

有一个长度为 N(1<=N<=100)的数列,并有 M(1<=M<=100)组约束关系。每一组约束关系包含四个参数: Ai,Bi,"gt"/"lt",Di,表示数列 S 中从 S[Ai]一直加到 S[Ai+Bi]比 Di 大("gt")或者比 Di 小("lt")。求问是否存在满足这 M 组约束关系的数列。

输入:

测试数据有若干组。

每一组第一行是两个整数 N 和 M。N=0 时表示测试数据结束,后面没有 M 的值。接下来 M 行,每一行包含四个参数:前两个是整数 Ai 和 Bi,然后是"gt"或者"lt",最后是 Di。四个参数之间用空格分隔。

输出:

每个测试数据一行,如果存在满足约束的数组,则输出"lamentable kingdom",否则输出 "successful conspiracy"。

题解:

按篇首提到的方法来构图进行差分约束系统的最短路求解。构造的边只有两种:原始式: s[a+b]-s[a-1]>ki,构图边:从[a+b]向[a-1]引一条权为(-ki-1)的边;原始式:s[a+b]-s[a-1]<ki,构图边:从[a-1]向[a+b]引一条权为ki-1的边。<span=""></<math>ki,构图边:从[a-1]向[a+b]引一条权为ki-1的边。<>

通过寻找负环来判断是否存在符合的解。这个时候用 SPFA 需要在一开始的时候让所有的点都入队列,避免某些点没有被检查到。这一操作可以有两种实现,一种是"超级源点",即虚构一个点,对所有其他点的边权为 0;第二种是手工入队,然后初始化 Dist 数组为 0。另外由于 0<=Ai+Bi<=100,所以有一个编号为 0 的点,一共有 n+1 个点,检查负环的时候需要是入队次数大于 n+1。

这题我做的时候卡了两个小时,都出在一些小问题上。其一是我在一组数据读完以后马上读了下一组的 N,结果干扰了整组的数据。其二是我 SPFA 枚举边的时候写了

nowpos:=First[now]; repeat...... nowpos:=Edges[nowpos].next; until nowpos=0;这样导致一开始 nowpos=0 的时候会进去循环体执行一次,而且数组越界了并没有报错! (嗯好像现在 FPC 都是这样的了)所以还是要改成 while (nowpos<>0) do 才行。这种小错误反映我写代码的时候会只顾着主体思路而忽视细节啊,一定要改.............

POJ1275【难】

题目大意:

超市需要一批出纳员,但是在每个时段需要的数目不同。超市经理希望雇佣最少的出纳员 以节省陈本。经理已经提供给你一天里每个小时需要的出纳员的最少数量: R0,

R1......R23。R0表示从午夜零时到午夜1时需要的出纳员的最少数目,R1表示从午夜1时到午夜2时需要的出纳员的最少数目,以此类推。每一天这些数据都是相同的。现在有N个人申请这个工作,每个申请者1在每24小时中,从某一时刻开始可以连续工作8个小时,然后无法工作,规定ti(0<=ti<=23)为他们开始工作的时刻。现在要求出最少雇佣多少个出纳员,满足每一个小时最少需要的出纳员人数。如果一个小时内工作的出纳员人数大于其必须值,也不会产生任何其他影响。

输入:

第一行一个整数,表示测试数据有多少组,这一数字将不大于20。

每一组测试数据中,第一行是 24 个被空格分隔的非负整数,依次代表 R0 到 R23,并且不超过 1000。接下来有一行,包含一个整数 N(0<=N<=1000)。再接下来有 N 行,每一行代表一个整数 ti。

两组测试数据之间没有任何空行进行分割。

输出:

对于每一组测试数据,输出一行:如果可以满足需求,则输出最少需要雇佣的出纳员数目:否则输出"No Solution"。

题解:

这题很难很经典,冯威的论文和《算法艺术与信息学竞赛》里面都有。难在建图的处理,以及需要枚举答案,后者一般是比较难想到的。另外冯威的标程里面对 n<>4 进行了特判,因为他漏加了一条边,详见 http://poj.org/showmessage?message_id=124693。另外这一道题的数组下标是挺坑爹的。

我写了 SPFA+模拟链表不知道为什么就是不对,明明建图已经和冯威的建法一样了但是就是不对,而 SPFA 部分我已经写了前面那么多题了也不可能出错……所以我放弃了,真是可耻。附上冯威修改后的标程。