最大流应用

最大流应用一

- POJ 2391 Ombrophobic Bovines
- 给定一个无向图,点 i 处有 Ai 头牛,点 i 处的牛棚能容纳 Bi 头牛,求一个最短时 间 T 使得在 T 时间内所有的牛都能进到某一牛棚里去。(1 <= N <= 200, 1 <= M <= 1500, 0 <= Ai <= 1000, 0 <= Bi <= 1000, 1 <= Dij <= 1,000,000,000)。

最大流应用一

- 本题是最大流相当基础的应用。做法比较简单。
- 将每个点 i 拆成两个点 i', i'',连边(s, i', Ai), (i'', t, Bi)。二分最短时间 T,若 d[i][j]<=T (d[i][j]表示点 i, j 之间的最短距离)则加边(i', j'', ∞)。每次根据最大流调整二分 的上下界即可。

最大流应用二

- POJ 1149 PIGS
- 有M个猪圈,每个猪圈里初始时有若干头猪。一开始所有猪圈都是关闭的。依次来了N个顾客,每个顾客分别会打开指定的几个猪圈,从中买若干头猪。每个顾客分别都有他能够买的数量的上限。每个顾客走后,他打开的那些猪圈中的 猪,都可以被任意地调换到其它开着的猪圈里,然后所有猪圈重新关上。问总共 最多能卖出多少头猪。(1<=N<=100,1<=M<=1000)举个例子来说。有3个猪圈,初始时分别有3、1和10头猪。依次来了3个顾客,第一个打开1号和2号猪圈,最多买2头;第二个打开1号和3号猪圈,最多买3头;第三个打开2号猪圈,最多买6头。那么,最好的可能性之一就是第一个 顾客从1号圈买2头,然后把1号圈剩下的1头放到2号圈;第二个顾客从3号圈买3头;第三个顾客从2号圈买2头。总共卖出2+3+2=7头。

最大流应用二

- 这个问题的网络模型可以很直观地构造出来，借上面的例子可以构造出以下模型：
- 三个顾客，就有三轮交易，每一轮分别都有 3 个猪圈和 1 个顾客的节点。
- 从源点到第一轮的各个猪圈各有一条边，容量就是各个猪圈里的猪的初始数量
- 从各个顾客到汇点各有一条边，容量就是各个顾客能买的数量上限。在某一轮中，从该顾客打开的所有猪圈都有一条边连向该顾客，容量都是 +∞。最后一轮除外，从每一轮的 i 号猪圈都有一条边连向下一轮的 i 号猪圈，容量都是 +∞，表示这一轮剩下的猪可以留到下一轮。
- 最后一轮除外，从每一轮被打开的所有猪圈，到下一轮的同样这些猪圈，两两之间都要连一条边，表示它们之间可以任意流通。

最大流应用二

- 这个网络模型的最大流量就是最多能卖出的数量。图中最多有 2+N+M×N≈100,000 个节点。这个模型虽然很直观，但是节点数太多了，计算速度肯定会很慢。
- 我们继续上面的例子，用合并的方法来简化这个网络模型
- 首先,最后一轮中没有打开的猪圈就可以从图中删掉了,也就是图2中红色的部分,显然它们对整个网络的流量没有任何影响。
- 如果几个节点的流量的**来源**完全相同，则可以把它们合并成一个
- 如果几个节点的流量的**去向**完全相同，则可以把它们合并成一个
- 如果从点 u 到点 v 有一条容量为 +∞ 的边，并且 u 是 v 的唯一流量来源，或者 v 是 u 的唯一流量去向，则可以把 u 和 v 合并成一个

最大流应用二

- 我们可以发现，合并完后，其实就成了下面这个模型。
- 每个顾客分别用一个节点来表示
- 对于每个猪圈的第一个顾客，从源点向他连一条边，容量就是该猪圈里的猪的初始数量。
- 如果从源点到一名顾客有多条边，则可以把它们合并成一条，容量相加。
- 对于每个猪圈，假设有 n 个顾客打开过它，则对所有整数 $i \in [1, n)$，从该猪圈的第 i 个顾客向第 $i + 1$ 个顾客连一条边，容量为 $+\infty$。
- 从各个顾客到汇点各有一条边，容量是各个顾客能买的数量上限。
- 可以这样理解这个新的网络模型：对于某一个顾客，如果他打开了猪圈 h，则在他走后，他打开的所有猪圈里剩下的猪都有可能被换到 h 中，因而这些猪都有可能被 h 的下一个顾客买走。所以对于一个顾客打开的所有猪圈，从该顾客到各猪圈的下一个顾客，都要连一条容量为 $+\infty$ 的边。

最大流应用三

- POJ 1637 Sightseeing tour
- 求混合图欧拉回路

最大流应用三

- 把该图的无向边随便定向,计算每个点的入度和出度。如果有某个点出入度之差为奇数,那么肯定不存在欧拉回路。因为欧拉回路要求每点入度 = 出度, 也就是总度数为偶数,存在奇数度点必不能有欧拉回路。好了,现在每个点入度和出度之差均为偶数。那么将这个偶数除以2,得x。也就是说,对于每一个点,只要将 x 条边改变方向(入>出就是变入,出>入就是变出),就能保证出=入。如果每个点都是出=入,那么很明显,该图就存在欧拉回路。现在的问题就变成了:我该改变哪些边,可以让每个点出=入?

最大流应用三

- 构造网络流模型。首先,有向边是不能改变方向的,我们不用再管。无向边定的是什么向,就把网络构建成什么样,边长容量上限 1。另新建 s 和 t。对于入>出的点u,连接边(u, t)、容量为x,对于出>入的点v,连接边(s, v), 容量为 x(注意对不同的点 x 不同)。之后,察看是否有满流的分配。有就是能有欧拉回路,没有就是没有。

- 欧拉回路是哪个?察看流值分配,将所有流量非0(上限是1,流值不是0就是1)的边反向,就能得到每点入度=出度的欧拉图。由于是满流,所以每个入>出的点,都有 x 条边进来,将这些进来的边反向, OK,入=出了。对于出>入的点亦然。那么,没和 s、t 连接的点怎么办?和 s 连 接的条件是出>入,和 t 连接的条件是入>出,那么这个既没和 s 也没和 t 连接的 点,自然早在开始就已经满足入=出了。那么在网络流过程中,这些点属于“中间点”。我们知道中间点流量不允许有累积的,这样,进去多少就出来多少,反 向之后,自然仍保持平衡。

最大流应用四

- POJ 2699 The Maximum Number of Strong Kings
- 一场联赛可以表示成一个完全图,点表示参赛选手,任意两点 u, v 之间有且仅有 一条有向边(u, v)或(v, u),表示 u 打败 v 或 v 打败 u。一个选手的得分等于被他打败的选手总数。一个选手被称为“strong king”当且仅当他打败了所有比他分高 的选手。分数最高的选手也是 strong king。现在给出某场联赛所有选手的得分序列,由低到高,问合理安排每场比赛的结果后最多能有几个 strong king。已知选手总数不超过 10 个。

最大流应用五

- POJ 3281 Dining
- 有 F 种食物和 D 种饮料,每种食物或饮料只能供一头牛享用,且每头牛只享用一 种食物和一种饮料。现在有 N 头牛,每头牛都有自己喜欢的食物种类列表和饮 料种类列表,问最多能使几头牛同时享用到自己喜欢的食物和饮料。(1 <= F <= 100, 1 <= D <= 100, 1 <= N <= 100)

最大流应用五

- 此题的建模方法比较有开创性。以往一般都是左边一个点集表示供应并与源相连，右边一个点集表示需求并与汇相连。现在不同了,供应有两种资源,需求仍只有 一个群体,怎么办?其实只要仔细思考一下最大流的建模原理,此题的构图也不 是那么难想。最大流的正确性依赖于它的每一条 s-t 流都与一种实际方案一一对 应。那么此题也需要用 s-t 流将一头牛和它喜欢的食物和饮料“串”起来,而食 物和饮料之间没有直接的关系,自然就想到把牛放在中间,两边是食物和饮料，由 s, t 将它们串起来构成一种分配方案。至此建模的方法也就很明显了:每种食 物 i 作为一个点并连边(s, i, 1),每种饮料 j 作为一个点并连边(j, t, 1),将每头牛 k 拆成两个点 k’, k”并连边(k’, k”, 1), (i, k’, 1), (k”, j, 1),其中 i, j 均是牛 k 喜欢的食物 或饮料。求一次最大流即为结果。

最大流应用六

- JOJ 2453 Candy　　　吉林大学acm题库
- 有 N 颗糖果和 M 个小孩,老师现在要把这 N 颗糖分给这 M 个小孩。每个小孩 i 对每颗糖 j 都有一个偏爱度 A_{ij},如果他喜欢这颗糖,$A_{ij} = 2$,否则 $A_{ij} = 1$。小孩 i 觉得高兴当且仅当$\sum C_{ij} \times A_{ij} >= B_i, j=1,2,\ldots,N$,若他分得了糖 j,$C_{ij} = 1$,否则 $C_{ij} = 0$。问能否合理分配这 N 颗糖,使得每个小孩都觉得高兴。($1 <= N <= 100,000$, $1 <= M <= 10$, $0<=B_i<=1,000,000,000$)

最大流应用六

- 一种最直观的想法就是每颗糖i作为一个点并连边(s, i, ?)，每个小孩j作为一个点并连边(j, t, Bj)。若小孩 j 喜欢糖果 i 则连边(i, j, 2),否则连边(i, j, 1),然后求一次最大流看是否等于ΣBj。但是问题也很明显,我们还没有给与源点关联的边确 定容量。实际上我们无法确定它们的容量,因为最大流无法实现这样一种控制: 一个点有若干条出边,容量不尽相同,现在要求经过该点的流可以任选一条出边流走,且一旦选定之后就只能从这条边流而不能再进入其他的出边。因此我们无 法确定与源关联的边的容量,因为经过每颗糖 i 的流无法在出边容量有1又有2的情况下作出正确的选择。

最大流应用六

- 那么是否就没有办法了呢?虽然流无法在容量有 1 又有 2 的情况下作出正确的选择,但却可以在容量有 1 又有 0 的情况下最自然地作出正确的选择,流过去就表 示选择了那条出边,且因为容量为 1,不会再有流进入其他的出边。那么此题的 构图方法也就出来了:每颗糖i作为一个点并连边(s, i, 1),每个小孩j作为一个点并连边(j, t, floor(Bj/2)),若小孩 j 喜欢糖果 i 则连边(i, j, 1),否则连边(i, j, 0)或者 干脆不连边,效果一样。设最大流为ans,若ans+N >=ΣBj则可以满足要求。 为什么?因为每颗糖迟早都要分给某个小孩,它一定会为总权值贡献 1,只不过 如果它分给了喜欢它的小孩就再额外贡献 1。现在我只考虑这额外的 1 单位贡献 究竟能送出去多少,最后加上基值 N 并与ΣBj 比较即可。

最大流应用七

- ZOJ 2760 How Many Shortest Path
- 给定一个带权有向图G=(V, E)和源点s、汇点t,问s-t边不相交最短路最多有几 条。(1 <= N <= 100)

最大流应用七

- 分别从源点和汇点作一次 Dijkstra,然后为建图方便仍保留所有的点,但只加入 满足 ds[u]+w[u][v]+dt[v]==dst的边(u, v)(这样便保证网络中的任意一条s-t路都 是最短路),容量为 1。求一次最大流即为结果。

- 注意不能按以下方法建图,因为这样会引进一些不在 s-t 最短路上的边: 保留所有满足 ds[u]+dt[u]==dst 的点 u,在这些点的导出子图中求最大流。原因自己思考,可以举出反例。

最大流应用八

- WOJ 1124 Football Coach   武汉大学acm题库
- 有N支球队,互相之间已经进行了一些比赛,还剩下M 场没有比。现在给出各支球队目前的总分以及还剩下哪M场没有比,问能否合理安排这M场比赛的结果,使得第N支球队最后的总分大于其他任何一支球队的总分。已知每场比赛胜者得2分,败者0分,平局则各得1分。(1<=N<=100, 0<=M<=1000)

最大流应用八

- 首先,贪心的让所有跟球队 N 相关的比赛都要球队N赢。如果此时仍有某支球队的总分大于等于球队N的总分,则已经不可能满足要求;否则按如下方法建图:每场比赛i(不包括与球队N相关的比赛)作为一个点并加边(s,i,2),每支球队j(不包括球队N)作为一个点并加边(j,t,score[N]-score[j]-1),每场比赛向与其关联的两支球队u,v连边(i,u,2),(i,v,2)。若最大流等于2*比赛场数(不包括与球队N相关的比赛)则可以满足要求。

最大流应用八

- 如果不允许平局存在,该如何构图?
- (s, i, 2) ---> (s, i, 1)
- (j,t,score[N]-score[j]-1)--->(j,t,floor((score[N]-score[j]-1)/2))
- (i, u, 2), (i, v, 2) ---> (i, u, 1), (i, v, 1)

最大流应用九

- SGU 326 Perspective http://acm.sgu.ru
- NBA 某小组内有 N 支球队,小组内以及小组间已经进行了若干场比赛。现在给出这 N 支球队目前胜利的场数、还剩多少场没有比(包括小组内和小组间)以及小组内任意两支球队之间还剩多少场没有比,问能否合理安排剩下的所有比赛,使得球队 1 最后胜利的场数大于等于小组内任何一支其他球队一样。(2<=N<=20,0<=x<=10000,x表示其他任何输入)

最大流应用九

- 此题和上一题非常相似。同样,所有和球队 1 相关的比赛全让球队 1 赢,如果此 时仍有某支球队胜利的场数大于球队 1,则已经不可能满足要求。按如下方法建 图:所有小组内的比赛i(不包括与球队1相关的比赛)作为一个点并加边(s, i, num[i]),每支球队(不包括球队1)作为一个点并加边(j, t, wins[1]-wins[i]),每场 比赛向与其关联的两支球队 u, v 连边(i, u, ∞), (i, v, ∞)。至于其他球队小组间的 比赛,直接让他们输掉就好,不用管。若最大流等于Σnum[i]则可以满足要求。

最大流应用十

- SGU 438 The Glorious Karlutka River =)
- 有一条东西向流淌的河,宽为 W,河中有 N 块石头,每块石头的坐标(Xi, Yi)和最大承受人数 Ci 已知。现在有 M 个游客在河的南岸,他们想穿越这条河流,但是每个人每次最远只能跳D米,每跳一次耗时 1 秒。问他们能否全部穿越这条河流,如果能,最少需要多长时间。(0 <= N <= 50, 0 < M <= 50, 0 <= D <= 1000, 0 < W <= 1000, 0 < Xi < 1000, 0 < Yi < W, 0 <= Ci <= 1000)

最大流应用十

- 经典的动态流问题,在流量限制的基础上加入了时间限制。首先将南岸作为源点, 北岸作为汇点,将每块石头拆成两个点(i', i'')并连边(i', i'', Ci)。接下来我们再按时间将每块石头拆点,形成一个分层网络time-expanded network,每块石头在不同的时间点都有一个点表示。所有能从南岸跳到的石头,从源点向其各时间点处连边,容量为∞。所有能跳到北岸的石头,从其各时间点处向汇点连边,容量为∞。 任意两块距离小于等于D的石头,互相从t到t+1连边,容量为∞。枚举t,并不断地往网络中加点表示当前时刻的石头,直到最大流等于总人数为止,此时的t即为结果。

最大流应用十一

- SPOJ 287 Smart Network Administrator
- 一座村庄有 N 户人家。只有第一家可以连上互联网,其他人家要想上网必须拉 一根缆线通过若干条街道连到第一家。每一根完整的缆线只能有一种颜色。网管 有一个要求,各条街道内不同人家的缆线必须不同色,且总的不同颜色种数最小。 求在指定的 K 户人家都能上网的前下,最小的不同颜色种数。(1 <= N <= 500)

最大流应用十一

- 给出一个不加证明的结论：最大颜色数等于最大边容量数。
- 此题乍一看不太好考虑,不知道如何去控制同一街道内的缆线颜色各不相同。我们不妨先尝试着建立一个网络模型出来:以第一家作为汇点 t,K 户人家中的每一户 i 作为一个点并连边(s, i, 1),对每条街道(u, v),连边(u, v, c), (v, u, c),c=∞。这样求完最大流后每一条 s-t 流均对应一户人家的缆线,而各条街道内的流量表示有多少户人家的缆线同时穿过该街道,那么这个流量就是只考虑该条街道的时候最少的不同颜色种数。那么答案是否就是所有街道的不同颜色种数的最大值呢? 当然不是!最大流只保证总流量最大,而不会去管每条流具体该往哪儿流,所以这么做不一定能得到最优解。我们只要稍微修改这个模型就一定能保证得到最优 解:强制 c 等于某个值 limit,再对网络求最大流,如果等于K,说明用c种不同的颜色已经足够了;如果小于 K,说明c种还不够,还需要往高了调。同时此处 的单调性也很明显:c越高越容易满足要求。思路一下就豁然开朗了:二分c的值,如果足够了就往下调,否则往上调,直到找到足够和不足够的临界值为止。

最大流应用十二

- SPOJ 962 Intergalactic Map
- 在一个无向图中,一个人要从 A 点赶往 B 点,之后再赶往 C 点,且要求中途不 能多次经过同一个点。问是否存在这样的路线。(3 <= N <= 30011, 1 <= M <= 50011)

最大流应用十二

- 由于每个点只能走一次,似乎最短路之类的算法不能用,只有往网络流上靠。将 每个点 i 拆成两个点 i', i''并加边(i', i'', 1)就能轻易达到这个目的。起初我一直以 A 为源点思考,却怎么也想不出如何处理先后经过两个汇点的问题,直到灵光一现, 想到可以以 B 为源点,A、C 为汇点,看能否增广两次。

最大流应用小结

- 最大流的应用一般都比较直观，很多时候每条流可能都对应着一个方案或是代表了其他的什么。
- 同时注意掌握建图技巧类似于拆点什么的。

最小割应用

最小割应用一

- HOJ 2634 How to earn more 湖南大学acm题库
- 有 M 个项目和 N 个员工。做项目 i 可以获得 Ai 元,但是必须雇用若干个指定的 员工。雇用员工 j 需要花费 Bj 元,且一旦雇用,员工 j 可以参加多个项目的开发。 问经过合理的项目取舍,最多能挣多少钱。(1 <= M, N <= 100)

最小割应用一

- 注意到题目中加粗的两个字“必须”,此题是典型的“蕴含式最大获利问题”,套 用解决最大权闭合子图的建模方法即可解决。每个项目i作为一个点并连边(s, i, Ai),每名员工 j 作为一个点并连边(j, t, Bj),若项目 i 需要雇用员工 j 则连边(i, j, ∞)。 设最小割为 ans,那么ΣAi-ans 即为结果。

最小割应用一

- 发散思维:
- 若进一步要求找出做了哪些项目以及雇用了哪些员工,该如何做? 从源点开始作一次 DFS 并对扫过的点进行标记,那么被标记的项目和员工即为 所求。
- 如果因做项目 i 而雇用员工 j,则员工 j 只能参加项目 i 的开发,若要做另一 个需要雇用员工 j 的项目 k,则需要重新付给员工 j 费用 Bj。该如何求解? 由于这样一来各个项目之间在雇用员工的意义上是相互独立的,直接贪心即可。

最小割应用二

- HOJ 2713 Matrix1
- 一个 N*M 的网格,每个单元都有一块价值 Cij 的宝石。问最多能取多少价值的宝石且任意两块宝石不相邻。(1 <= N, M <= 50, 0 <= Cij <= 40000)

最小割应用二

- 经典的最大点权独立集问题。转化为最小点权覆盖集问题:先将网格黑白染色,从源点到每个黑点有一条边,从每个白点到汇点有一条边,容量均为相应宝石的价值。 每个黑点向与其相邻的四个白点连边,容量为∞。设最小割为 ans,结果即为Σ Cij – ans。

最小割应用三

- POJ 1815 Friendship
- 现代社会人们都靠电话通信。A 与 B 能通信当且仅当 A 知道 B 的电话号或者 A 知道C的电话号且C与B能通信。若A知道B的电话号,那么B也知道A的电 话号。然而不好的事情总是会发生在某些人身上,比如他的电话本丢了,同时他又换了电话号,导致他跟所有人失去了联系。现在给定 N 个人之间的通信关系 以及特定的两个人S和T,问最少几个人发生不好的事情可以导致S与T无法通信并输出这些人。如果存在多组解,输出字典序最小的一组。(2 <= N <= 200)

最小割应用三

- 此题是求一个最小点割集,但是要求输出方案且要字典序最小。首先仍是拆点构图,求一次最小割记为 ans。之后我们采取贪心的策略,从 1 至 N 枚举删点,如 果最小割不变,说明该点不可能在最小割中,我们再把该点加入到网络中;否则 最小割一定变小了,用这个较小值更新 ans,记录该点是一个解并不再将其放回。 重复这个过程,把所有点都扫一遍后结果就出来了。

最小割应用四

- ZOJ 2532 Internship
- 有 N 个城市,M 个中转站以及 L 条有向边(u, v, c),表示可以从 u 向 v 传送信息，带宽为 c。每个城市都在向 CIA 总部发送无穷大的信息量,但是目前总部实际接 收带宽已经不能满足要求。CIA 决定要增大某条边的带宽以增大总部的接收带宽，请找出哪些边带宽的增加能导致总部接收带宽的增加。(1 <= N+M <= 100, 1 <= L <= 1000)

最小割应用四

- 此题要求找出这样一条边,增加它的容量可以导致最大流的增加。最直观的做法是先求一次最大流记为ans,然后枚举每条边的容量加1再求最大流,看是否大于ans。但是这样做复杂度太高。有没有更好的办法?如果我们换个角度,考虑求完最大流后残量网络R中的割,那么问题便迎刃而解。最大流的含义是我们现在在R中找不到一条从s走到t的增广路。我们只要找到这样的割边(u, v),使得R中从s能走到u,从v能走到t,那么当我增加这条割边的容量后,它就会再次 出现在 R 中,架起一条增广路 s-u-v-t,从而增加最大流的流量。再正式述一下 算法流程:求一次最大流,然后在残量网络R中以正向弧对s、以正向弧的逆对t作 DFS,并用 from[i], to[i]标记点 i 能否从 s 可达,能否可达 t。扫每一条正向 弧(u, v),若它残留容量为 0 且 from[u]且 to[v],则(u, v)为一个解。

- 发散思维:如何判定网络的最小割是否唯一?同样求一次最大流后在残量网络 R 中以正向弧对 s、以正向弧的逆对 t 作 DFS, 只不过这次只用一个 flag[i]标记点 i 是否在两次 DFS 中被探访过。最后扫一遍所 有点,如果存在没有被探访过的,则说明最小割不唯一。

最小割应用五

- Ural 1277 Cops and Thieves http://acm.timus.ru
- 一个犯罪团伙打算去偷一家美术馆。警察决定派 K 个人堵住所有从匪窝通向美术 馆的道路,不过他们只能驻守在沿途顶点处而不能在匪窝或美术馆,且每个点都 有一个需要警察驻守的最低人数 Ri。问警察能否完成任务。(2 < N <= 100, 1 < M <= 10000)

最小割应用五

- 将每个点 i 拆成两个点 i’, i”,除匪窝 s 和 美术馆 t 外加边(i’, i”, Ri),将每条无向边(i, j)分解为(i”, j’), (j”, i’)。令 s”为源,t’为 汇,求一次最小割即为结果。(注意此题还需要特判匪窝和美术馆在同一点的情况)

最小割应用六

- SPOJ 839 Optimal Marks
- 给出一个无向图,每个点有一个标号 mark[i],不同点可能有相同的标号。对于 一条边(u, v),它的权值定义为 mark[u] xor mark[v]。现在一些点的标号已定,请 决定剩下点的标号,使得总的边权和最小。(0 < N <= 500, 0 <= M <= 3000, 0 <= mark[i] <= 2^31-1)

最小割应用六

- 先规范化问题,不妨设所有点一定和已知标号的点连通。否则,令那些不与已知 标号点连通的点的标号为 0,这些点间的边权也不必计入目标函数(边权都为 0)。 在问题的优化式子中,发现异或操作不好处理,难以转化为一些基本运算。深入 分析异或操作的本质我们会发现,各个二进制位间是互不影响的。所以我们可以 分别处理各个二进制位,最后再汇总即可。处理每个二进制位时构图如下:保留 原始标号,将每个点的最优标号置 0;对每个已经有原始标号的点 i,若 i 的该位 是 1 则连边(s, i, ∞),否则连边(i, t, ∞);对原图中的每条边(i, j),连边(i, j, 1), (j, i, 1)。求一次最小割后,对源点DFS,所有被探访到的点,将其最优标号中的该位置1即可。

最小割应用七

- SPOJ 1693 Coconuts
- N 个城堡守卫正在就非洲的燕子能否搬运椰子而进行投票。每个人都有自己的看法,但是为了避免跟自己的朋友持相反意见,他们 时常会投相反的票。现在给出每个人的初始看法以及朋友关系,求在某种投票方 案下,违背自己意愿的票数与持不同意见的朋友对数的总和最小。(2 <= N <= 300, 1 <= M <= N(N-1)/2)。

最小割应用七

- 此题是经典的“二者取一问题”。每名守卫 i 作为一个点,若他投赞成票, 则加边(s, i, 1), (i, t, 0),否则加边(s, i, 0), (i, t, 1)(设最小割中与源 s 连通的点表示 赞同);若 i 跟 j 是朋友,则加边(i, j, 1), (j, i, 1)。求一次最小割即为结果。

- 发散思维:如何求这个函数的最小值?$E(x_1, x_2, \ldots, x_n) = \Sigma(1-x_i)*|p_i-v_0| + \Sigma x_i*|p_i-v_1| + \Sigma|x_i-x_j|*|p_i-p_j|$, p_i,v_0,v_1均为常数,$x_i \in \{0, 1\}$

- 不管这个函数表达什么意思,光是分析它的结构我们就能发现这是一个“二者取其一式问题”:把前两项合起来考虑就会发现,要么 $x_i=0$ 并获得一个$|p_i-v_0|$, 要么 $x_i=1$并获得一个$|p_i-v_1|$;考虑第三项,如果 x_i, x_j 取不同值时获得一个$|p_i-p_j|$。 与这个模型惊人的相似!构图思路和之前一模一样,求一次最小割即为结果。

最小割应用小结

- 最小割是最大流的对偶问题,但在实际建模过程中,它不如最大流来的直观,模型也往往隐蔽得很深,不容易找到构图方法。以下是几种常见的出题模式：
- (1) 用 s-t 割集表示方案这是最小割最基本的构图法,每一个 s-t 割集都实实在在地对应着实际问题中的一种操作方案,比其他应用好理解。
- (2) 蕴含式最大获利问题。就是那个最大闭合子图。
- (3) 二者取其一式问题。常考。这类建模方法也有一个明显的特点,就是每个点都可以有两种方案供选择,每种方案都有一个花费,必须且只能选择其中一 种;另外如果某两个点选择了不同的方案,还要在这两个点之间增加额外的费用。 这种应用其实也可算作第一种构图法,只不过它的特点更明显,所以我把它单列出来。有的时候还可以与数据结构结合，有兴趣的同学可以去做一下bzoj的a+b problem。