网络流建模选讲

祖赫良　星之羊驼#5729

PART1 最小割

能不能一句话解决大部分最小割问题

能不能一句话解决大部分最小割问题

VFleaKing

NOI金牌 12

如果v向u连一条容量为w的有向边，表示v如果在S割，那么u不在S割会产生w的代价。
一个等价的表述是，u如果在T割，那么v不在T割会产生w的代价。注意v如果在T割，那么u在S割是不会产生代价的。
特别的，如果v向u连一条容量为正无穷大的有向边，表示v如果在S割，那么u一定也要在S割。
一个等价的表述是，u如果在T割，那么v一定也要在T割。

具体是单向边还是双向边取决于你要实现的功能。

SHOI2007 善意的投票

n个小朋友通过投票来决定睡不睡午觉。每个人都有自己的主见，但也可以投和自己本来意愿相反的票。冲突总数为好朋友之间发生冲突的总数加上和自己本来意愿发生冲突的人数。求最小冲突数。

SHOI2007 善意的投票

- 把小朋友按照自己意愿分成两组,这样构成了一个二分图。假设S割表示选1,T割表示选0这样只需要
- 源点向每个选1的小朋友连一条容量为1的边,表示如果这个小朋友没在S割中会产生1的代价。
- 每个选0的小朋友向汇点连一条容量为1的边,表示如果这个小朋友没在T割中会产生1的代价。
- 每对好朋友之间互相连一条容量为1的边,表示如果他们分别在两个割中会产生1的代价。

NOI2006 最大获利

一共有n个基站,建设第i个基站的代价为Pi
一共有m种获利方式,Ai Bi Ci表示如果第Ai个基站和第Bi个基站都建立了的话则可以收入Ci,求最大获利

NOI2006 最大获利

因为获利=收入-代价,我们需要求的就是使代价最小,我们先假设获得了全部的收入,这样接下来就容易算代价

假设S割表示被选择了,T割表示没有被选择

做法一：

为每条边建一个新的结点Ci(注意与收入的Ci区分),S向这个结点连容量为Ci的边表示如果没在S割会造成Ci的代价(也就是相当于收入的逆运算),此外Ci向这条边所连接的两个点Ai和Bi连一条容量为无穷大的边,表示如果选了这条边则两侧的点也必须要选。

原来的点向T连一条容量为Pi的边表示如果在S割会造成Pi的代价

NOI2006 最大获利

做法二：
并不建立新的结点,对于每条边,S向Ai和Bi分别连一条容量为Ci的边,用途见下面
然后Ai和Bi相互连一条容量为Ci的边,表示如果Ai和Bi不在同一个割中则会造成Ci的代价
那么我们讨论一下Ai与Bi在不同割中的影响
如果Ai在S割,Bi在T割,则会造成2*Ci的代价,反之亦然
如果Ai和Bi都在S割,则没有代价
如果Ai和Bi都在T割则会造成2*Ci的代价
于是结果就是如果Ai和Bi都选了则没代价,其他情况都造成2*Ci的代价
所以只需要每个点向T连一条容量为2*Pi的边,这样算出来的代价就正好是两倍,除以二即可。

HDU4307 Matrix

- 给你一个n*n的矩阵B和1*n的矩阵C,求一个1*n的01矩阵A,使得
- D=(AB-C)AT最大,AT为A的转置

HDU4307 Matrix

- D的运算式子是这样的

$$D = \sum_{i=1}^{N} \left(\left(\sum_{j=1}^{N} a_j \cdot b_{ji} \right) - c_i \right) \cdot a_i$$

HDU4307 Matrix

- 展开可以得到

$$D=\sum_{i=1}^{N}\sum_{j=1}^{N}a_i\cdot a_j\cdot b_{ij}-\sum_{i=1}^{N}a_i\cdot c_i$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{N}b_{ij}-\sum_{i=1}^{N}\sum_{j=1}^{N}(1-a_i\cdot a_j)\cdot b_{ij}-\sum_{i=1}^{N}a_i\cdot c_i$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{N}b_{ij}-(\sum_{\substack{i=1\\a_i=0}}^{N}\sum_{j=1}^{N}b_{ij}+\sum_{\substack{i=1\\a_i=1}}^{N}\sum_{\substack{j=1\\a_j=0}}^{N}b_{ij}+\sum_{\substack{i=1\\a_i=1}}^{N}a_i\cdot c_i)$$

HDU4307 Matrix

- 展开可以得到

$$D=\sum_{i=1}^{N}\sum_{j=1}^{N}a_i\cdot a_j\cdot b_{ij}-\sum_{i=1}^{N}a_i\cdot c_i$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{N}b_{ij}-\sum_{i=1}^{N}\sum_{j=1}^{N}(1-a_i\cdot a_j)\cdot b_{ij}-\sum_{i=1}^{N}a_i\cdot c_i$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{N}b_{ij}-\left(\sum_{\substack{i=1\\a_i=0}}^{N}\sum_{j=1}^{N}b_{ij}+\sum_{\substack{i=1\\a_i=1}}^{N}\sum_{\substack{j=1\\a_j=0}}^{N}b_{ij}+\sum_{\substack{i=1\\a_i=1}}^{N}a_i\cdot c_i\right)$$

如果把bij看成选i到j这条边的收入,ai看成是否选第i个点,ci看成选第i个点的代价,是不是很像最大获利?

HDU4307 Matrix

- 与最大获利不同的是最大获利双向边只记一次收入,这里记两次
- 所以就可以直接按照最大获利的做法二来建模,并且因为计算两次收入,所以我们也不需要把最大流除以2什么的了
- 但是注意此题边数多达10^6所以我们不能像最大获利里那样任性地每条边都从S向两个点连一条新边而是像下面那样求和
- 不考虑最大获利直接建模的话就是一共n个点来表示ai的状态,S向第i个点连容量为∑bij的边表示如果ai=0,也就是不在S割,会造成∑bij的代价,第i个点向第j个点连容量为bij的边表示ai=1且aj=0会造成bij的代价,第i个点向T连容量为ci的边表示如果ai=1会造成ci的代价

HDU5572 String Problem

- 给你一个长度为n的只包含0-9的字符串选出其中任意位
- 再给你一个n*n的矩阵B表示如果选出了第i位与第j位,则可以获得Bij+Bji的收益
- 此外每个数字(0-9)给你一个系数ai与bi如果选出的数字i的个数ki>0的话则会产生ai(ki-1)+bi的代价
- 求最大收益

HDU5572 String Problem

- 如果不考虑数字的代价的话,这道题跟上一道题一模一样,所以关键是如何处理每个数字的代价
- 数字i的代价的公式可以改写为ai*ki+(bi-ai)
- 这样我们每个数字建立一个新的结点,每个原来的点根据字符串上的数字向新的点连一条容量为无穷的边,表示选了某个字符的结点则必须选择其对应数字的结点,再向T连一条容量为ai的边,表示如果选了那个点会造成ai的代价
- 新建的结点向T连一条容量为bi-ai的边,表示选了这个数字会造成bi-ai的代价,但是bi-ai<0怎么办?

HDU5535 Chess Puzzle

- 给你一个n*m的棋盘,需要往格子上面放黑色或白色(B||W)的棋子
- 有些格子开始已经放了某种棋子,其他的格子可以任意放直到放满为止
- 放满之后对于棋盘上的i和j两个格子,如果|xi-xj|=a且|yi-yj|=b(类似于国际象棋中的马)的话可以得1分
- 求最大得分以及字典序最小的放法

HDU5535 Chess Puzzle

- 把每个格子与它对应的格子连边然后黑白染色变成二分图(注意与棋子的黑白不同),以下简称为左边的点和右边的点
- 于是问题变成了如果左边与右边的点颜色不同的话可以得1分,预先统计分数的话变成了如果左边与右边的点颜色相同的话会造成1分的代价,怎么看上去有点不对?

HDU5535 Chess Puzzle

- 实际上我们发现左边的点的颜色跟右边点的颜色没有任何关系,所以我们把右边点的颜色黑白颠倒(注意初始已经存在的点的颜色也应对应颠倒),于是问题就变成了如果左边点的颜色与右边点不同的话会造成1的代价,这样正常连边就好了,初始存在的点就根据其颜色来决定是从S向它连容量为无穷大的边还是从它向T连容量为无穷大的边
- 这样得分求完了,如何求字典序最小的方案?

HDU5535 Chess Puzzle

- 只需要从左上角开始枚举点的颜色,如果B不可行的话就是W,然后根据残量网络来决定与其相连的点的颜色,这样每个格子最多访问两次,所以复杂度并不高

XTUOJ1248 TC or CF

- 一共有n场比赛,每场题目风格是TC或CF的
- 一共有m个参赛者,第i个人先参加第ai场比赛然后参加第bi场比赛(比赛是平行的所以有可能存在ai==bj且bi==aj)
- 如果第i个人先参加TC然后参加CF的话会获得ci的怒气值
- 所以如何安排每场比赛的风格使得至少有3场TC且至少有3场CF且第一场必是TC最后一场必是CF还能让怒气值最少

XTUOJ1248 TC or CF

- 其实建模很简单,一共n个点,每个ai向bi连一条容量为ci的边
- S向第一个点连容量为正无穷的边
- 最后一个点向T连容量为正无穷的边
- 如何保证其他的比赛至少有两场TC与两场CF?
- n^4枚举=TLE

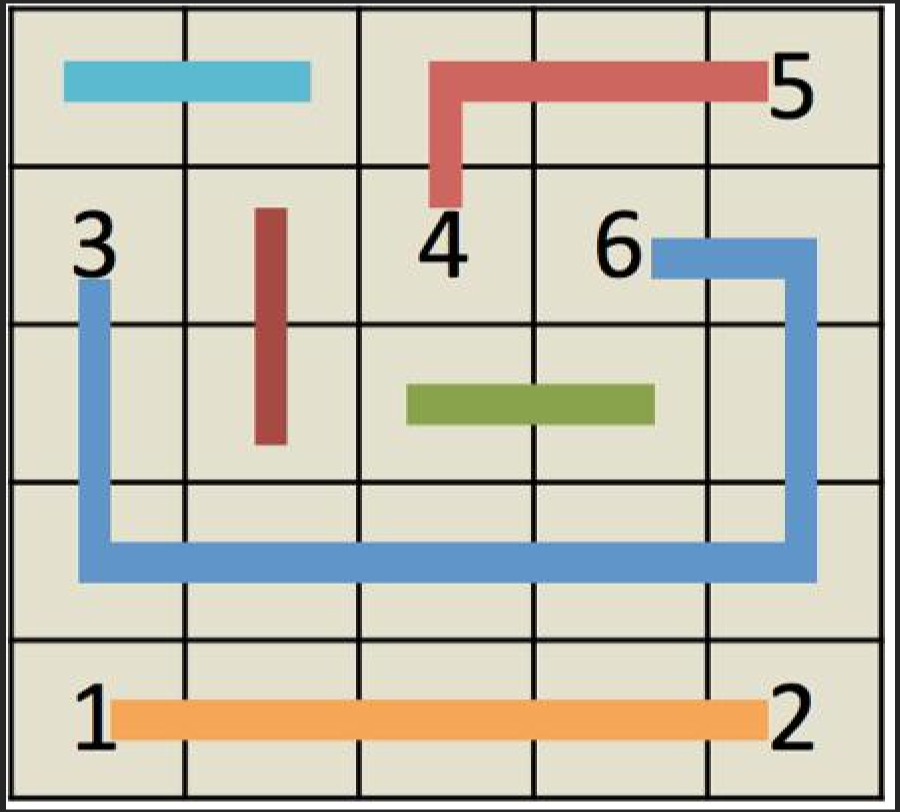
XTUOJ1248 TC or CF

- 割的排中律：一个点只可能在S割或T割而不会是两者皆非的
- 所以枚举第二个点和第三个点在哪个割即可
- 如果都在S割,则只需枚举在T割的点
- 如果都在T割,则只需枚举在S割的点
- 如果一个在S割一个在T割,则只需枚举一个在S割的点和一个在T割的点
- 复杂度4*n^2

PART2 网络流与环

HDU5520 Number Link

- 给你一个n*m的棋盘,有些上面是数字,其他是空格,在上面连很多不相交的线使得奇数一定连向偶数,其他的都是环或1*2(看成是两个格子组成的环)连接相邻两个格子的费用已给出,问是否可行与最小费用

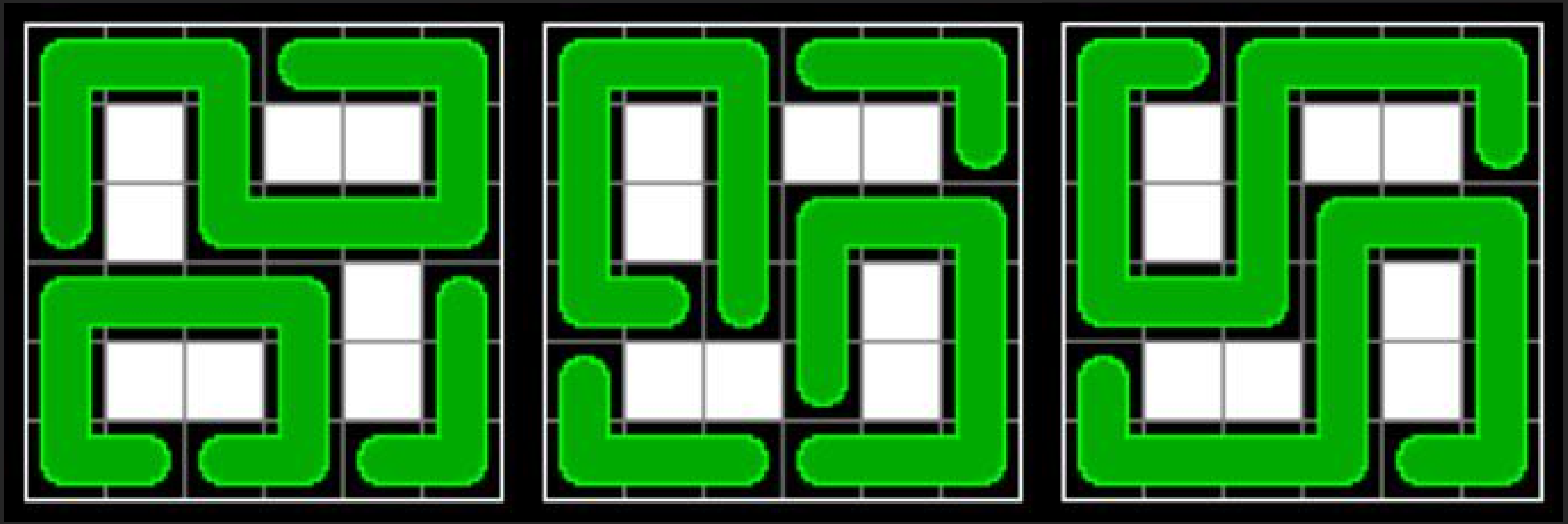


HDU5520 Number Link

- 每个格子一定在某条连线线上,假设是从奇数向偶数连线,则每个奇数点入度为0出度为1,每个偶数点入度为1出度为0,空格入度为1出度为1
- 每个点拆成两个点分别表示入点和出点,S向每个空格与奇数的入点连一条容量为1费用为0的边,每个空格和偶数的出点向T连一条容量为1费用为0的边,当它们满流时分别表示那个点的入度和出度为1
- 每个入点向其相邻的点连容量为1费用为所给费用的边表示连线
- 最小费用最大流就是结果,如果奇数偶数个数不想等或流量小于空格+奇数的数量就说明无解

BZOJ4213 Snake

- 给你一个n*m的棋盘,上面有一些障碍物,其他的格子上都被一些蛇占据且它们的头与尾要么都在边界上(长度>=2),要么首尾相接成一个环(长度>=4),求头尾在边界的蛇的数量最少是多少



BZOJ4213 Snake

- 对棋盘黑白染色,分为左右两个部分的点,每个环形蛇的头尾必定在两侧。
- 从S向左侧每个点连容量为2的边,从右侧每个点向T连容量为2的边来表示每个点的入度和出度,从左侧每个点向相邻的右侧点连容量为1的边,这样如果能跑满流则说明每个点入度和出度都是1这个图可以用环来填满,
- 那么对于边界上的点来说入度和出度至少有1个是1即可,所以只需设定流量下界为1,而其他点流量下界等于上界等于2,只需要求出这样一个满足流量上下界的最大流然后统计有多少个点流量为1除以2就是答案,其实也可以改写成最小费用可行流,这样求出的费用除以二也是答案
- 此题模型能否与上题互换?