

## POJ2914 【基础】

题目大意：

给出一个有权无向图，求其最小割。

输入：

有若干组测试数据。每一组测试数据的前两行有两个整数  $N$  和  $M$ ，表示有  $N$  个点（依次编号为 1 到  $N$ ）和  $M$  条边（ $2 \leq N \leq 500$ ， $1 \leq M \leq (N-1)*N/2$ ）。接下来有  $M$  行，每一行有三个整数  $A$ 、 $B$  和  $C$ ，表示  $A$  和  $B$  之间有一条容量为  $C$  的边。两组测试数据之间没有空行。

输出：

对于每一组测试数据，输出一行，包含一个整数，即其最小割的。如果图不连通，输出 0。

题解：

这题我用来做了我的 Stoer-Wagner 算法模板，是从

[http://blog.sina.com.cn/s/blog\\_6635898a0100qwd0.html](http://blog.sina.com.cn/s/blog_6635898a0100qwd0.html) 的第一个模板改过来的。顺带一提，这篇文章后面的那个模板跑了 8000+ms，第一个模板跑了 3100+ms……

## POJ2125 【基础】

题目大意：

给出一张图，有  $N$  个顶点和  $M$  条有向边。每一次可以取走一个点和它的所有出边，或者一个点和它的所有入边。对于  $i$  点，取走所有入边的代价为  $W_i^+$ ，取走所有出边的代价为  $W_i^-$ 。求删除图中所有边的最小代价。

输入：

输入的第一行包含两个整数  $N$  和  $M$ （ $1 \leq N \leq 100$ ， $1 \leq M \leq 5000$ ）。第二行有  $N$  个整数，依次为  $W_1^+$  到  $W_N^+$ 。第三行有  $N$  个整数，依次为  $W_1^-$  到  $W_N^-$ 。接下来有  $M$  行，每一行有两个整数  $u$ 、 $v$ ，表示有一条有向边从  $u$  指向  $v$ 。

输出：

第一行输出一个整数  $W$ ，即删除图中所有边的最小代价。第二行输出一个整数  $K$ ，即需要进行多少次操作。接下来输出  $K$  行，每一行先输出一个整数，表示要操作的点，后面用一个空格分开再输出一个 “+” 或者 “-” 号，表示是删掉该点的入边还是出边。

题解：

这题也算是一个入门题了，是最小点权覆盖，推导过程在胡伯涛的论文《最小割模型在信息学竞赛中的应用》的 5.4，这里我不再赘述。需要注意的是，最后按残余流量搜索结果的时候，直接从源开始搜，在  $1 \sim N$  中不被搜到的点表示拆除出边的，在  $N+ (1 \sim N)$  中被搜到的点作出拆除入边的。因为源点到  $1 \sim N$  的点的边残余流量为 0 即意味着取该点出边的操作被执行了，而  $N+ (1 \sim N)$  到汇点的边容量为 0 即意味着取该店入边的操作被执行了。

## POJ1966 【基础】

题目大意：

给出一个无向图  $G$ ，有  $N$  个顶点 ( $1 \leq N \leq 100$ )，规定其点连通度为  $K(G)$ ，如果在  $G$  中去掉任意  $K-1$  个点  $G$  仍然连通但去掉任意  $K$  个点时  $G$  不连通。如果图里的各点都是孤立的，那么规定  $K(G) = 0$ 。求一个图的  $K(G)$ 。

输入：

有若干组测试数据，每一组测试数据占一行。每一行一开始有两个整数  $N$  和  $M$ ，表示有  $N$  个顶点和  $M$  条边。接下来有  $M$  组格式为  $(u_i, v_i)$  的数据，表示点  $u_i$  和  $v_i$  之间有一条边。两组边数据之间有一个空格分隔。

输出：

每一组测试数据输出一行，即  $K(G)$ 。

题解：

这题是一个最小点割集，可以转化为最小割最大流来做，转化原理和独立轨有关。所谓独立轨，即两条从源点到汇点的路径，没有公共边。因此，一个有向图的最小割边数量就等于其独立轨，就等于其最大流。所以，求最小点割集，也就是要把点转换成边，来求最小割边，因此需要对每一个点  $u$  拆点成  $u'$  和  $u''$ ，并且  $u'$  到  $u''$  的容量为 1。具体转化方式如下：

若  $G$  是无向图，将  $G$  的无向边拆成两条有向边，然后  $G$  是有向图；

如果  $G$  是有向图：

- 1、将  $G$  里的每一个顶点  $n$  拆点拆成  $n'$  和  $n''$ ，从  $n'$  向  $n''$  连一条有向边，容量为 1
  - 2、将原图中每一条有向边  $u \rightarrow v$  变成  $u' \rightarrow u'' \rightarrow v' \rightarrow v''$ ，其中  $u'' \rightarrow v'$  的容量为正无穷
  - 3、某一点  $A''$  是源点， $B'$  是汇点。
- 求解的时候先指定一点  $A''$  作为源点，然后枚举汇点，求最大流里的最小值。所有具有流量 1 的有向边  $(v', v'')$  对应的  $v$  顶点组成一个割顶集，在  $G$  中去掉这些顶点则  $G$  变成不连通。若求出最大流的值大于等于点数，则表示这是一个强连通图，不存在割点。

## P0J3469 【基础】

题目大意：

某计算机有两个处理器。现在有  $N$  个 ( $1 \leq N \leq 20000$ ) 模块工作需要计算机进行处理，已知第  $i$  个模块工作在两个处理器上的耗时分别是  $A_i$  和  $B_i$  (我吐个槽，难道这俩核心是不同架构的？要不然的话所有条件一样怎么可能出现这种情况。)，各模块工作没有先后顺序。另外，有  $M$  对 ( $1 \leq M \leq 200000$ ) 模块之间可能需要共用数据，而如果这两个模块在同一个处理器上工作，那么就不需要额外的时间开销，如果分别在两个处理器上工作那么就需要额外的时间开销。现在请求出完成这些模块所需要的最少的时间。

输入：

第一行有两个整数  $N$  和  $M$ 。

接下来有  $N$  行，每一行有两个整数  $A_i$  和  $B_i$ ，表示第  $i$  个模块在两个处理器上面的耗时。

接下来有  $M$  行，每一行有三个整数  $a_i$ ， $b_i$  和  $w_i$ ，表示  $a_i$  和  $b_i$  模块如果在两个处理器上工作，需要有  $w_i$  的额外耗时。

输出：

一行，一个整数，即最小的总耗时。

题解：

这题还是可以用最小割的模型来解决。将两个处理器作为源点和汇点，每一个模块作为一个点，该模块在两个处理器上的耗时作为该点的一条入边和一条出

边，而所有协同工作的关系则在两个点之间连两条单向边，权值为额外耗时。然后转化为最大流来做。下面说明一下为什么这么建图是合理的：

当 AB 之间没有边连通的时候，显然割边为点 A 与 CPU1 和 CPU2 连接中耗费最小的一个，对于 B 也是一样。但是当 A 与 B 之间有边时，若最小割集 S 中包含 A，割集 T 中包含 B，很显然最小割边必然有 AB 边，若割边不包含 AB，则源点汇点还是连通的，那就不为割。所以构图时若 A 和 B 有协作的额外耗费，就将 AB 连接，边权为消耗值。

另外值得一提的是，这一题同样的 ISAP 代码，用 G++ 交比用 GCC 交要多 250ms 左右，看来有时候 C 比 C++ 的确跑得快一点。这里我给出了我的 C++ 和 C 代码，不相信的话诸君可以亲自测试一下。

## POJ3084 【基础】

题目大意：

有若干个房间，这些房间之间有一些门。这些门连接两个房间，但是关上以后只能从其中一个房间来打开。现在所有的门都是开着的。有若干个恐怖分子在其中的某一些房间里，我们要保护某一间房间，需要知道最少需要关闭多少个房门，才能使恐怖分子不能直接进入被保护的房间。

输入：

文件的第一行有一个整数 X，表示有 X 组测试数据。

每一组测试数据的第一行有两个整数 m ( $1 \leq m \leq 20$ ) 和 n ( $0 \leq n \leq 19$ )，m 表示房间数量，n 表示要被保护的房间编号（房间从 0 到 m-1 依次编号）。接下来有 m 行，第 i 行开头有两个或一个英文字母 NI 或者 I，NI 表示第 i 行中没有恐怖分子，I 表示第 i 行中有恐怖分子。接下来有一个整数 c ( $0 \leq c \leq 20$ )，后面有 c 个整数  $r_1 \dots r_c$ ，表示这个房间可以打开 c 扇门，分别通往  $r_1 \dots r_c$  号房间。

两组测试数据之间没有空行。

输出：

每一组测试数据输出一行，即最少需要关闭的门的数量。如果无论如何都无法确保指定房间的安全，输出“PANIC ROOM BREACH”。

题解：

这题是最小割。由于一扇门“正向”总可以通过，“反向”的话只有在门开了的时候才能通过，所以我们建两条边，正向边容量无穷，反向边容量为 1，源和恐怖分子所在的房间相连，汇就是被保护的房间。由于我们需要关闭最少的

门，也就是找到最少的割边，所以直接跑一次最大流就好了。当发现最大流为 INF 的时候，直接返回无法确保安全即可。

## POJ2987 【基础】

题目大意：

现在你是一个公司的老总，为了挽救公司你必须裁掉一些人。你手上有一张多年积累下来的调查表，知道裁掉每一个人的收益是多少（可能为负数）。但是，公司里山头林立，如果你裁掉了某一个人，那么他将会带着他的下属，他的下属会带着下属的下属……直到最底层的员工一起离开。所以，你需要计算出最少需要裁掉多少人，并且能获得多少收益。

输入：

输入的第一行有两个整数  $N$  和  $M$ ，表示公司除了你以外有  $N$  ( $0 \leq N \leq 5000$ ) 人依次编号为 1 到  $N$ ，和已知  $M$  ( $0 \leq M \leq 60000$ ) 对员工之间的上下属关系。接下来有  $N$  行，每一行有一个整数  $W_i$  ( $|W_i| \leq 10^7$ )，表示裁掉第  $i$  个员工的收益是  $W_i$ 。接下来有  $M$  行，每一行有两个整数  $u$  和  $v$ ，表示编号为  $u$  的员工是编号为  $v$  的员工的上司。

输出：

一行，包含两个空格分隔的整数，即最少需要裁掉的人数和对于的最大收益。

题解：

这题是典型的最大权闭合图，相关的内容在胡伯涛的论文里有，我不再赘述。

需要提及的一点是，最大收益需要用 long long (C/C++) 或者 Int64

(Pascal) 才能保存下来（因为最大可能是  $6 \times 10^{11} > 2.1 \times 10^{10}$ ），我在这里挂了一次。另外这次 C 跑得没有 C++ 快，似乎 ISAP 跑得没有 Dinic 快（后者我没有验证过）。

## POJ3308 【中等】

题目大意：

有  $L$  ( $1 \leq L \leq 500$ ) 个外星人将会在一个  $N \times M$  的区域内同时降落 ( $1 \leq N, M \leq 50$ )，我们需要在它们落地的时候同时消灭他们。现在我们有一种地图炮 SLB，每一次可以将一行或者一列内的外星人全部消灭。SLB 安装在不同的行或者列有不同的代价，而总的代价和是所有安装了 SLB 的行列的代价之积。请求出最小的总代价。

输入：

第一行有一个整数，表示输入包含多少组测试数据。

每一组测试数据的第一行有三个整数，分别是  $N$ 、 $M$  和  $L$ 。接下来有两行，第一行有  $N$  个小数，第二行有  $M$  个小数，第  $i$  个小数表示装在第  $i$  行/列的代价。接下来有  $L$  行，每一行有两个整数，表示一个外星人降落的行列坐标。

输出：

对于每一组测试数据，输出一个小数，保留到小数点后 4 位，即最小的总代价。

题解：

这题其实还是最小点权覆盖，虽然最后的代价是乘积，但是只要先取对数加和后再取指数就好了。用类似 POJ3041 的方式进行行和列的匹配，每一个点就是二分图的一条边，然后按胡伯涛论文里介绍的方法去做就好了。另外似乎需要注意一下精度问题。

这里有一个比胡伯涛的论文要好理解一点的介绍最小点权覆盖的文章：

<http://blog.csdn.net/lyy289065406/article/details/6784658>

## POJ1815 【中等】

题目大意：

当代社会的社交网络中，人们往往通过一对一的方式来进行交流。我们规定，如果两个人互相交换了手机号码，那么他们可以相互联系。另外，如果  $A$  和  $B$  交换了手机号码， $B$  和  $C$  也交换了手机号码，我们可以推断  $A$  和  $C$  也可以相互联系。但是有时候因为一些因素，有些人保存的所有手机号码会丢失或者失效，而相应地他原来可以联系的那些人也联系不到他了。

现在，有  $N$  个 ( $2 \leq N \leq 200$ ) 人依次从 1 到  $N$  进行编号，你知道他们之间谁和谁能够互相联系。然后给出两个人的编号  $S$  和  $T$ ，你要求出最少有多少人出现事故的时候可以导致  $S$  和  $T$  之间不能互相联系，并且出事故的人按序号升序排列要满足字典序最小。

输入：

第一行有三个整数  $N$ 、 $S$  和  $T$  ( $1 \leq S \neq T \leq N$ )。接下来有  $N$  行每一行有  $N$  个整数，第  $i$  行第  $j$  个整数是 1/0 表示编号为  $i$  的人知道/不知道编号为  $j$  的人的手机号。

输出：

如果  $S$  和  $T$  总能保持联系，输出一行 “NO ANSWER!”。

否则，第一行输出一个整数  $t$ ，表示最少需要有多少个人发生意外才能使  $S$  和  $T$  之间失去联系。第二行输出  $t$  个整数，即发生意外的那几个人，按字典序升序输出。

题解：

此题和 POJ1966 的原理一样都是求最小点割集，不过已经给出了源点和汇点。最蛋疼的是要求割点的字典序最小。我想了一下似乎没有什么很好的方法只能枚举了，即在跑了一遍最大流以后从标号为 1 的点开始枚举（源汇点除外），构图时把当前点忽略掉，如果除去这个点能令最大流减一的话那么就将这个点标记。但是我犯了个很严重的错误，那就是原本找到  $k$  个割点后找后面的第  $k+1$  个割点的时候构图应该把前面的  $k$  个割点也给忽略掉，但是我的做法是每次只忽略当前寻找的第  $k+1$  个割点，然后看看出来的最大流是不是原来的最大流-1。这里存在一个很严重的问题就是如果两个点在一条独立轨上那么按后者的做法可能同时被标记，但是不应该同时被标记的。还好在 Discuss 里看到一组数据让我意识到了这个问题。唉，解题思维还是不够严谨啊。

## POJ3155 【难】

题目大意：

John 是某家公司的 CEO，某一天公司董事会董事长的儿子 Scott 进入了公司管理层，董事长让 John 给 Scott 一个团队来管理。John 担心 Scott 做得太好会抢了自己的位置，于是决定挑选出一个尽量难管的团队给 Scott。John 经过长期的积累知道公司里每两个人之间在同一个团队中的合作是否有困难，而一个团队的合作难度系数就是这个团队中所有人中合作有困难的组合数除以人数。现在他要挑选出一个合作难度系数最大的团队来给 Scott。

输入：

第一行有两个整数  $n$  和  $m$  ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 1000$ )，表示公司有  $n$  个人（从 1 依次编号到  $n$ ），其中有  $m$  对人互相合作有困难。接下来有  $m$  行，每一行有两个整数  $a$ 、 $b$ ，表示编号为  $a$  和  $b$  的两个人合作有困难。

输出：

第一行输出一个整数  $k$ ，表示挑选出来的团队有多少人。接下来按升序输出这  $k$  个人编号，每一个编号一行。如果有多个解，输出任意一个即可。

题解：

这题是裸的一个最大密度子图问题，在胡伯涛的论文里有详解，不过我看起来感觉比较费劲，上网找了个：

[http://blog.sina.com.cn/s/blog\\_64675f5401001bmh.html](http://blog.sina.com.cn/s/blog_64675f5401001bmh.html) 对照着看了一下，很艰难地写了代码还是没过。回头还要再仔细啃论文才行啊。