Graph Matching

Hany01

Yali High School

May 24, 2018

匈牙利算法

匈牙利算法

非常暴力的做法。

匈牙利算法

非常暴力的做法。

每次匹配的时候，如果当前点i没有被匹配，那么直接匹配i；如果当前点i被mat_i匹配过了，那么递归处理检查mat_i是否可以匹配到其他点即可。

匈牙利算法

非常暴力的做法。
每次匹配的时候，如果当前点i没有被匹配，那么直接匹配i；如果当前点i被mat_i匹配过了，那么递归处理检查mat_i是否可以匹配到其他点即可。
时间复杂度$O(nm)$

网络流解法

网络流解法

超级源点连向所有S集合的点连一条边，S集合与T集合间按题目要求连边，T集合的所有点向超级汇点连边。（流量均为1）

网络流解法

超级源点连向所有S集合的点连一条边，S集合与T集合间按题目要求连边，T集合的所有点向超级汇点连边。（流量均为1）
直接跑网络流即可，时间复杂度逊于匈牙利。

网络流解法

超级源点连向所有S集合的点连一条边，S集合与T集合间按题目要求连边，T集合的所有点向超级汇点连边。（流量均为1）
直接跑网络流即可，时间复杂度逊于匈牙利。
~~但是网络流可以套板子啊233~~

性质

性质

一些概念:

性质

一些概念：

最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。

性质

一些概念：

最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。

最小边覆盖：每个点至少连一条边。

性质

一些概念：
最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。
最小边覆盖：每个点至少连一条边。
最大独立集：集合中的任何两个点都不直接相连。

性质

一些概念：

最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。

最小边覆盖：每个点至少连一条边。

最大独立集：集合中的任何两个点都不直接相连。

最小点覆盖=最大匹配数

性质

一些概念：

最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。

最小边覆盖：每个点至少连一条边。

最大独立集：集合中的任何两个点都不直接相连。

最小点覆盖=最大匹配数

最小边覆盖=二分图点数-最大匹配数

性质

一些概念：

最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。

最小边覆盖：每个点至少连一条边。

最大独立集：集合中的任何两个点都不直接相连。

最小点覆盖=最大匹配数

最小边覆盖=二分图点数-最大匹配数

最大独立集=总数-最小点覆盖

性质

一些概念：

最小点覆盖：即在所有顶点中选择最少的顶点来覆盖所有的边。

最小边覆盖：每个点至少连一条边。

最大独立集：集合中的任何两个点都不直接相连。

最小点覆盖=最大匹配数

最小边覆盖=二分图点数-最大匹配数

最大独立集=总数-最小点覆盖

二分图的最大独立集等于其补图的最大团

一些性质的伪证

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。
那么每一条边只有两种情况：

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。
那么每一条边只有两种情况：
1. 一端为红，一端为蓝

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。
那么每一条边只有两种情况：

1. 一端为红，一端为蓝
2. 两端都为红

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。
那么每一条边只有两种情况：
1. 一端为红，一端为蓝
2. 两端都为红

不可能出现两端都为蓝色，想一想，为什么。

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。
那么每一条边只有两种情况：
1. 一端为红，一端为蓝
2. 两端都为红
不可能出现两端都为蓝色，想一想，为什么。
然后用所有一端为蓝点的边的另一端作为最小覆盖集中的点即可，发现刚好是在每一对匹配中取一个点。

一些性质的伪证

将图中被匹配的点染成红色，未被匹配的点染成蓝色。
那么每一条边只有两种情况：
1. 一端为红，一端为蓝
2. 两端都为红
不可能出现两端都为蓝色，想一想，为什么。
然后用所有一端为蓝点的边的另一端作为最小覆盖集中的点即可，发现刚好是在每一对匹配中取一个点。
所以最小覆盖数等于最大匹配数。

一些性质的伪证

一些性质的伪证

把最小覆盖集中的点去掉后，剩下的点显然是最大独立集。

一些性质的伪证

把最小覆盖集中的点去掉后，剩下的点显然是最大独立集。
所以最大独立集与最小覆盖集互为补集，即总数-最小覆盖数=最大独立集。

一些性质的伪证

把最小覆盖集中的点去掉后，剩下的点显然是最大独立集。
所以最大独立集与最小覆盖集互为补集，即总数-最小覆盖数=最大独立集。
最大匹配后，每个未匹配的点连出一条边，即为最小边覆盖=二分图点数-2*最大匹配+最大匹配=二分图点数-最大匹配。

一些性质的伪证

把最小覆盖集中的点去掉后，剩下的点显然是最大独立集。
所以最大独立集与最小覆盖集互为补集，即总数-最小覆盖数=最大独立集。
最大匹配后，每个未匹配的点连出一条边，即为最小边覆盖=二分图点数-2*最大匹配+最大匹配=二分图点数-最大匹配。
参考博客

一些性质的伪证

把最小覆盖集中的点去掉后，剩下的点显然是最大独立集。
所以最大独立集与最小覆盖集互为补集，即总数-最小覆盖数=最大独立集。
最大匹配后，每个未匹配的点连出一条边，即为最小边覆盖=二分图点数-2*最大匹配+最大匹配=二分图点数-最大匹配。
参考博客
等到dyx讲课时会解释得更详细，这里就不再赘述了。

Hall定理

Hall定理

- 对于二分图$G = (S \cup T, E)$，设T中与S'中的点有边相连的点集为$\Gamma(S')$，它存在完美匹配当且仅当$S' \subseteq S$, $|S'| \leq |\Gamma(S')|$

Hall定理

- 对于二分图$G = (S \cup T, E)$，设T中与S'中的点有边相连的点集为$\Gamma(S')$，它存在完美匹配当且仅当$S' \subseteq S$, $|S'| \leq |\Gamma(S')|$
- 拓展：二分图的最大匹配数等于$|S| - \max(|S'| - |\Gamma(S')|)$

[TJOI2013]攻击装置

给定一个01矩阵，其中你可以在0的位置放置攻击装置。 每一个攻击装置(x, y)都可以按照“日”字攻击其周围的8个位置$(x-1, y-2), (x-2, y-1), (x+1, y-2), (x+2, y-1), (x-1, y+2), (x-2, y+1), (x+1, y+2), (x+2, y+1)$

[TJOI2013]攻击装置

给定一个01矩阵，其中你可以在0的位置放置攻击装置。 每一个攻击装置(x, y)都可以按照“日”字攻击其周围的8个位置$(x-1, y-2), (x-2, y-1), (x+1, y-2), (x+2, y-1), (x-1, y+2), (x-2, y+1), (x+1, y+2), (x+2, y+1)$
求在装置互不攻击的情况下，最多可以放置多少个装置。

[TJOI2013]攻击装置

给定一个01矩阵，其中你可以在0的位置放置攻击装置。 每一个攻击装置(x, y)都可以按照“日”字攻击其周围的8个位置$(x-1, y-2), (x-2, y-1), (x+1, y-2), (x+2, y-1), (x-1, y+2), (x-2, y+1), (x+1, y+2), (x+2, y+1)$
求在装置互不攻击的情况下，最多可以放置多少个装置。
$N \leq 200$

[TJOI2013]攻击装置

[TJOI2013]攻击装置

对图进行黑白染色，从而将图分成两个集合，可以互相攻击到的点连边，求最大独立集即可。

[BZOJ3546][ONTAK2010]Life of the Party

给定一个二分图最大匹配，求出所有关键点。

[BZOJ3546][ONTAK2010]Life of the Party

3.2.1 二分图最大匹配关键点

关键点是指的一定在最大匹配中的点。

由于二分图左右两侧是对称的，我们只考虑找左侧的关键点。

先求任意一个最大匹配 M，要找的关键点此时一定都是匹配点。考虑 M 中的一个匹配点 p，设 M' 为某个不包含 p 的最大匹配，对称差 $D = M \oplus M'$，则 D 中一定存在一条以 p 为端点的偶交替链，这条链另一端不在 M 中。

那么一个匹配点 s 能变成非匹配点，当且仅当从这个点出发能找一条以匹配边出发的交替链，使得终点是某个未盖点 t。由于链长为偶数，t 和 s 属于同一侧（左侧）。

我们倒过来考虑，先给二分图定向：匹配边从右到左、非匹配边从左到右，从左侧每个未盖点出发 DFS，给到达那些点打上标记。最终左侧每个没有标记的匹配点即为关键点。因为只关心可达性，显然每个点只需访问至多一次，复杂度 $O(n+m)$。

[BZOJ3546][ONTAK2010]Life of the Party

3.2.1 二分图最大匹配关键点

关键点是指的一定在最大匹配中的点。

由于二分图左右两侧是对称的，我们只考虑找左侧的关键点。

先求任意一个最大匹配 M，要找的关键点此时一定都是匹配点。考虑 M 中的一个匹配点 p，设 M' 为某个不包含 p 的最大匹配，对称差 $D = M \oplus M'$，则 D 中一定存在一条以 p 为端点的偶交替链，这条链另一端不在 M 中。

那么一个匹配点 s 能变成非匹配点，当且仅当从这个点出发能找一条以匹配边出发的交替链，使得终点是某个未盖点 t。由于链长为偶数，t 和 s 属于同一侧（左侧）。

我们倒过来考虑，先给二分图定向：匹配边从右到左、非匹配边从左到右，从左侧每个未盖点出发 DFS，给到达那些点打上标记。最终左侧每个没有标记的匹配点即为关键点。因为只关心可达性，显然每个点只需访问至多一次，复杂度 $O(n+m)$。

（摘自2015年中国国家队候选队员论文集 陈胤伯《浅谈图的匹配算法及其应用》）

[HEOI2016/TJOI2016]游戏

Description

[HEOI2016/TJOI2016]游戏

[HEOI2016/TJOI2016]游戏

我们对于每一行、每一列以硬石头为界分为很多个块。

[HEOI2016/TJOI2016]游戏

我们对于每一行、每一列以硬石头为界分为很多个块。
那么对于每一个块都最多只能放一个炸弹。

[HEOI2016/TJOI2016]游戏

我们对于每一行、每一列以硬石头为界分为很多个块。
那么对于每一个块都最多只能放一个炸弹。
所以就可以对于每一个空地，将其所在横向块、纵向块连边，跑匈牙利就行了。

[HEOI2016/TJOI2016]游戏

我们对于每一行、每一列以硬石头为界分为很多个块。
那么对于每一个块都最多只能放一个炸弹。
所以就可以对于每一个空地，将其所在横向块、纵向块连边，跑匈牙利就行了。

[网络流24题]最小路径覆盖问题

Description

[网络流24题]最小路径覆盖问题

- 这道题并不难，只是让大家都了解一下套路。

[网络流24题]最小路径覆盖问题

- 这道题并不难，只是让大家都了解一下套路。
- 建一个二分图，如果原DAG上有u连向v，那么二分图上u连向v'

[网络流24题]最小路径覆盖问题

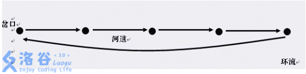
- 这道题并不难，只是让大家都了解一下套路。
- 建一个二分图，如果原DAG上有u连向v，那么二分图上u连向v'
- 一开始每个点都被看作一条独立的路径，每连一条边就减少一条路径，所以答案就是点数减去最大匹配数。

[网络流24题]最小路径覆盖问题

- 这道题并不难，只是让大家都了解一下套路。
- 建一个二分图，如果原DAG上有u连向v，那么二分图上u连向v'
- 一开始每个点都被看作一条独立的路径，每连一条边就减少一条路径，所以答案就是点数减去最大匹配数。
- 代码

[BZOJ1143][CTSC2008]祭祀river

在遥远的东方，有一个神秘的民族，自称Y族。他们世代居住在水面上，奉龙王为神。每逢重大庆典，Y族都会在水面上举办盛大的祭祀活动。我们可以把Y族居住地水系看成一个由岔口和河道组成的网络。每条河道连接着两个岔口，并且水在河道内按照一个固定的方向流动。显然，水系中不会有环流（下图描述一个环流的例子）。



由于人数众多的原因，Y族的祭祀活动会在多个岔口上同时举行。出于对龙王的尊重，这些祭祀地点的选择必 须非常慎重。准确地说，Y族人认为，如果水流可以从一个祭祀点流到另外一个祭祀点，那么祭祀就会失去它神圣的意义。族长希望在保持祭祀神圣性的基础上，选择尽可能多的祭祀的地点。

[BZOJ1143][CTSC2008]祭祀river

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念:

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念:

- 链：集合中任意两个点u,v，要么u可以走到v，要么v可以走到u

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念：

- 链：集合中任意两个点u,v，要么u可以走到v，要么v可以走到u
- 反链：任意两个点谁也不能走到谁。

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念:

- 链：集合中任意两个点u,v，要么u可以走到v，要么v可以走到u
- 反链：任意两个点谁也不能走到谁。
- 最长反链：就是反链中最长的那个。

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念：

- 链：集合中任意两个点u,v，要么u可以走到v，要么v可以走到u
- 反链：任意两个点谁也不能走到谁。
- 最长反链：就是反链中最长的那个。

很显然最长反链等于最小路径覆盖啊。

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念：

- 链：集合中任意两个点u,v，要么u可以走到v，要么v可以走到u
- 反链：任意两个点谁也不能走到谁。
- 最长反链：就是反链中最长的那个。

很显然最长反链等于最小路径覆盖啊。
证明：orz vfleaking

[BZOJ1143][CTSC2008]祭祀river

其实这道题和上面那个最小路径覆盖本质上是一样的！！
概念:

- 链：集合中任意两个点u,v，要么u可以走到v，要么v可以走到u
- 反链：任意两个点谁也不能走到谁。
- 最长反链：就是反链中最长的那个。

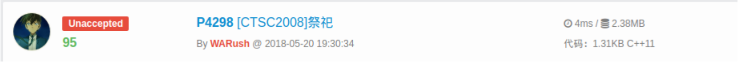
很显然最长反链等于最小路径覆盖啊。
证明：orz vfleaking
这道题就这么做完了?

[BZOJ1143][CTSC2008]祭祀river

还是Sometimes Naive了XD

[BZOJ1143][CTSC2008]祭祀river

还是Sometimes Naive了XD



[BZOJ1143][CTSC2008]祭祀river

为什么会这样?

[BZOJ1143][CTSC2008]祭祀river

为什么会这样?
这里要区分两个概念:

[BZOJ1143][CTSC2008]祭祀river

为什么会这样?
这里要区分两个概念:

- 最小不相交路径覆盖:可以用上面的方法求出。

[BZOJ1143][CTSC2008]祭祀river

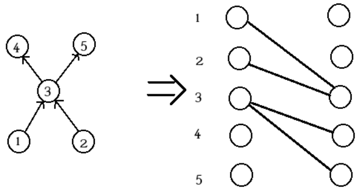
为什么会这样?
这里要区分两个概念:

- 最小不相交路径覆盖:可以用上面的方法求出。
- 最小相交路径覆盖:需要先传递闭包,再进行匹配!!

[BZOJ1143][CTSC2008]祭祀river

为什么会这样?
这里要区分两个概念:

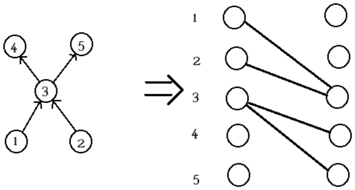
- 最小不相交路径覆盖：可以用上面的方法求出。
- 最小相交路径覆盖：需要先传递闭包，再进行匹配！！



[BZOJ1143][CTSC2008]祭祀river

为什么会这样?
这里要区分两个概念:

- 最小不相交路径覆盖:可以用上面的方法求出。
- 最小相交路径覆盖:需要先传递闭包,再进行匹配!!



然后会发现其实不就是求出是否可以抵达后、求最大独立集么233

[HNOI2013]消毒

小T的分格实验皿是一个长方体，被划分为$a \times b \times c$个单位立方体区域。小T被导师要求将其中一些单位立方体区域进 行消毒操作。他被要求使用一种特定的F试剂来进行消毒。这种F试剂特别奇怪，每次对尺寸为$x \times y \times z$的长方体区域进行消毒时，只需要使用$\min(x, y, z)$单位的F试剂。现在请你告诉他，最少要用多少单位的F试剂。（输入保证$a \times b \times c \leq 5000$）

[HNOI2013]消毒

小T的分格实验皿是一个长方体，被划分为$a \times b \times c$个单位立方体区域。小T被导师要求将其中一些单位立方体区域进 行消毒操作。他被要求使用一种特定的F试剂来进行消毒。这种F试剂特别奇怪，每次对尺寸为$x \times y \times z$的长方体区域进行消毒时，只需要使用$\min(x, y, z)$单位的F试剂。现在请你告诉他，最少要用多少单位的F试剂。（输入保证$a \times b \times c \leq 5000$）

Hint: 可以先考虑二维的情况。

[HNOI2013]消毒

由于当我们选择一个区域时，可以对除最小维度之外的两个维度任意扩展，所以我们可以只考虑每次消毒都是$x \times y \times 1$的形式。

[HNOI2013]消毒

由于当我们选择一个区域时，可以对除最小维度之外的两个维度任意扩展，所以我们可以只考虑每次消毒都是$x \times y \times 1$的形式。
因为$abc \leq 5000$，那么最小的一维最大也只有17，那么枚举那一个维度有哪些层被直接消毒了。

[HNOI2013]消毒

由于当我们选择一个区域时，可以对除最小维度之外的两个维度任意扩展，所以我们可以只考虑每次消毒都是$x \times y \times 1$的形式。
因为$abc \le 5000$，那么最小的一维最大也只有17，那么枚举那一个维度有哪些层被直接消毒了。
至于剩下的，我们从另外两个维度进行消毒，就变成了经典的二维上的最小覆盖问题辣！

[HEOI2012]朋友圈

Description

[HEOI2012]朋友圈

啥？一般图的最大团？？怕莫不是要提前挑战NPC了。

[HEOI2012]朋友圈

啥？一般图的最大团？？怕莫不是要提前挑战NPC了。
先只看A国人：只有两个数一奇一偶时才能成为朋友，那么它们的关系构成了一个二分图，二分图的最大团为2。

[HEOI2012]朋友圈

啥？一般图的最大团？？怕莫不是要提前挑战NPC了。
先只看A国人：只有两个数一奇一偶时才能成为朋友，那么它们的关系构成了一个二分图，二分图的最大团为2。
再看B国人：奇数和奇数、偶数和偶数是两两相连的，一些奇数和一些偶数有边相连，发现它们关系的补图构成了一个二分图！而二分图的最大独立集等于其补图的最大团！！

[HEOI2012]朋友圈

啥？一般图的最大团？？怕莫不是要提前挑战NPC了。
先只看A国人：只有两个数一奇一偶时才能成为朋友，那么它们的关系构成了一个二分图，二分图的最大团为2。
再看B国人：奇数和奇数、偶数和偶数是两两相连的，一些奇数和一些偶数有边相连，发现它们关系的补图构成了一个二分图！而二分图的最大独立集等于其补图的最大团！！
所以我们先枚举在A国人中选哪两个人，然后拿出与选出的人相连的B国人，并建出补图，求出最大独立集即可。

[BZOJ4950][WF2017] Mission Improbable

Description

[BZOJ4950][WF2017] Mission Improbable

- 考虑满足的条件:

[BZOJ4950][WF2017] Mission Improbable

- 考虑满足的条件:
- 1. 对于俯视图：之前该方格内有箱子，那么最终至少要剩一个。

[BZOJ4950][WF2017] Mission Improbable

- 考虑满足的条件：
- 1. 对于俯视图：之前该方格内有箱子，那么最终至少要剩一个。
- 2. 对于每一行每一列的最高的那一个不能动。

[BZOJ4950][WF2017] Mission Improbable

- 考虑满足的条件:
- 1. 对于俯视图：之前该方格内有箱子，那么最终至少要剩一个。
- 2. 对于每一行每一列的最高的那一个不能动。
- 我们考虑先将所有的箱子抽掉并保留每一行、每一列最高的格子，然后考虑哪些格子可以同时作为一行、一列的最高格子，这个显然可以用二分图匹配。

[BZOJ4950][WF2017] Mission Improbable

- 考虑满足的条件：
- 1. 对于俯视图：之前该方格内有箱子，那么最终至少要剩一个。
- 2. 对于每一行每一列的最高的那一个不能动。
- 我们考虑先将所有的箱子抽掉并保留每一行、每一列最高的格子，然后考虑哪些格子可以同时作为一行、一列的最高格子，这个显然可以用二分图匹配。
- 注意：不同的最大值不会互相干扰，所以可以直接匹配。

[Codeforces 720A] Closing Ceremony

Description

有一个$n \times m$ 的网格，每个位置上有一个椅子。有k个人在$(0,0)$，有l个人在$(0,m+1)$，每个人有一个耐力值，能否给每个人都安排一个椅子，使得每个人从起始位置到椅子的曼哈顿距离不超过他的耐力值。

[Codeforces 720A] Closing Ceremony

Description

有一个$n \times m$ 的网格, 每个位置上有一个椅子。有k个人在$(0, 0)$, 有l个人在$(0, m+1)$, 每个人有一个耐力值, 能否给每个人都安排一个椅子, 使得每个人从起始位置到椅子的曼哈顿距离不超过他的耐力值。

Hint: Hall定理

[Codeforces 720A] Closing Ceremony

~~傻逼贪心？？~~

[Codeforces 720A] Closing Ceremony

~~傻逼贪心？？~~

- 我们令人为S，椅子为T，对于每一个$S' \subseteq S$，$\Gamma(S')$一定是所有满足到$(0,0)$距离小于i，到$(0,m+1)$距离小于j的椅子。

[Codeforces 720A] Closing Ceremony

~~傻逼贪心？？~~

- 我们令人为S，椅子为T，对于每一个$S' \subseteq S$，$\Gamma(S')$一定是所有满足到$(0,0)$距离小于i，到$(0,m+1)$距离小于j的椅子。
- 其中i为S'中位于$(0,0)$的点中的最大体力值，j为S'中位于$(0,m+1)$的点中的最大体力值。

[Codeforces 720A] Closing Ceremony

~~傻逼贪心？？~~

- 我们令人为S，椅子为T，对于每一个$S' \subseteq S$，$\Gamma(S')$一定是所有满足到$(0,0)$距离小于i，到$(0,m+1)$距离小于j的椅子。
- 其中i为S'中位于$(0,0)$的点中的最大体力值，j为S'中位于$(0,m+1)$的点中的最大体力值。
- 所以对于每一个$\Gamma(S')$，我们求出最大的$|S'|$即可。

[Codeforces 720A] Closing Ceremony

~~傻逼贪心？？~~

- 我们令人为S，椅子为T，对于每一个$S' \subseteq S$，$\Gamma(S')$一定是所有满足到$(0,0)$距离小于i，到$(0,m+1)$距离小于j的椅子。
- 其中i为S'中位于$(0,0)$的点中的最大体力值，j为S'中位于$(0,m+1)$的点中的最大体力值。
- 所以对于每一个$\Gamma(S')$，我们求出最大的$|S'|$即可。
- 枚举i,j，最大的$|S'|$即为所有在$(0,0)$体力值不小于i、在$(0,m+1)$体力值不小于j的人。

[Codeforces 720A] Closing Ceremony

~~傻逼贪心？？~~

- 我们令人为S，椅子为T，对于每一个$S' \subseteq S$，$\Gamma(S')$一定是所有满足到$(0,0)$距离小于i，到$(0,m+1)$距离小于j的椅子。
- 其中i为S'中位于$(0,0)$的点中的最大体力值，j为S'中位于$(0,m+1)$的点中的最大体力值。
- 所以对于每一个$\Gamma(S')$，我们求出最大的$|S'|$即可。
- 枚举i,j，最大的$|S'|$即为所有在$(0,0)$体力值不小于i、在$(0,m+1)$体力值不小于j的人。
- 求出$\max(|S'|-|\Gamma(S')|)$即可

KM算法

KM算法

其实和匈牙利有些类似，只是引入了一个顶标的概念。

KM算法

其实和匈牙利有些类似，只是引入了一个顶标的概念。

定义一个点x的顶标为$L(x)$，那么对于任意边(u, v)满足$L(u) + L(v) \geq w(u, v)$。

KM算法

其实和匈牙利有些类似，只是引入了一个顶标的概念。
定义一个点x的顶标为$L(x)$，那么对于任意边(u,v)满足$L(u)+L(v)\geq w(u,v)$。
先把S中的点的顶标都赋成该点可以连接的边中的最大边权，T中的点的顶标均为0。

KM算法

其实和匈牙利有些类似，只是引入了一个顶标的概念。
定义一个点x的顶标为$L(x)$，那么对于任意边(u,v)满足$L(u)+L(v)\geq w(u,v)$。
先把S中的点的顶标都赋成该点可以连接的边中的最大边权，T中的点的顶标均为0。
每次求出最小的顶标修改量更新顶标值，其实就是扩大可匹配点的范围，直到不能再修改即可。

KM算法

其实和匈牙利有些类似，只是引入了一个顶标的概念。
定义一个点x的顶标为$L(x)$，那么对于任意边(u,v)满足$L(u)+L(v)\geq w(u,v)$。
先把S中的点的顶标都赋成该点可以连接的边中的最大边权，T中的点的顶标均为0。
每次求出最小的顶标修改量更新顶标值，其实就是扩大可匹配点的范围，直到不能再修改即可。
时间复杂度$O(n^4)$，可以存一下最小修改量，变成$O(n^3)$。

KM算法

其实和匈牙利有些类似，只是引入了一个顶标的概念。
定义一个点x的顶标为$L(x)$，那么对于任意边(u,v)满足$L(u)+L(v)\geq w(u,v)$。
先把S中的点的顶标都赋成该点可以连接的边中的最大边权，T中的点的顶标均为0。
每次求出最小的顶标修改量更新顶标值，其实就是扩大可匹配点的范围，直到不能再修改即可。
时间复杂度$O(n^4)$，可以存一下最小修改量，变成$O(n^3)$。
一个很有趣的关于KM算法的博客

费用流解法

在二分图匹配的网络流解法的基础上加上费用即可。

模板题们

[POJ3565] Ants
[HDU2255] 奔小康赚大钱
[POJ2195] Going Home
[TJOI2014] 匹配 (得到最大匹配结果后尝试删掉每一条匹配边然后再跑KM即可)
[BZOJ2589][CTSC2000] 丘比特的烦恼

模板题们

[POJ3565] Ants
[HDU2255] 奔小康赚大钱
[POJ2195] Going Home
[TJOI2014] 匹配 (得到最大匹配结果后尝试删掉每一条匹配边然后再跑KM即可)
[BZOJ2589][CTSC2000] 丘比特的烦恼
坑点

[POJ3686] The Windy's

有N个订单和M个机器，给出第i个订单在第j个机器完成的时间$M[i,j]$，每台机器同一时刻只能处理一个订单，机器必须完整地完成一个订单后才能接着完成下一个订单。

[POJ3686] The Windy's

有N个订单和M个机器，给出第i个订单在第j个机器完成的时间$M[i, j]$，每台机器同一时刻只能处理一个订单，机器必须完整地完成一个订单后才能接着完成下一个订单。
问N个订单完成时间的平均值最少为多少。

[POJ3686] The Windy's

比较经典的建模题。

[POJ3686] The Windy's

比较经典的建模题。
发现一个倒数第k处理的订单，时间被计算了k次。

[POJ3686] The Windy's

比较经典的建模题。
发现一个倒数第k处理的订单，时间被计算了k次。
考虑每个机器拆成n个点，表示是倒数第几个处理的。

[POJ3686] The Windy's

比较经典的建模题。
发现一个倒数第k处理的订单，时间被计算了k次。
考虑每个机器拆成n个点，表示是倒数第几个处理的。
对于订单i，机器j的第k个点，连接一条权值为$M[i,j] \times k$的边，然后跑KM即可。

[POJ3686] The Windy's

比较经典的建模题。
发现一个倒数第k处理的订单，时间被计算了k次。
考虑每个机器拆成n个点，表示是倒数第几个处理的。
对于订单i，机器j的第k个点，连接一条权值为$M[i,j] \times k$的边，然后跑KM即可。
Code

[HDU2282] Chocolate

有n个盒子组成一个圆，盒子里总共有不超过n个蛋糕，有的有好几个，有的为0。可以将一个盒子里的蛋糕往左右两个盒子里移动，一次只能移动一个，使最终每个盒子里有不超过一个蛋糕（可以没有），求最小的移动数。

[HDU2282] Chocolate

将有超过一个蛋糕的盒子作为一个集合（有k个蛋糕的盒子拆成$k-1$个点），没有蛋糕的盒子作为一个集合，之间的匹配边的权值为盒子之间的距离，跑KM即可。

[COCI2006-2007 #1] Bond

[COCI2006-2007 #1] Bond

有 n 个人去执行 n 个任务，每个人执行每个任务有不同的成功率，每个人只能执行一个任务，求所有任务都执行的总的成功率。

[COCI2006-2007 #1] Bond

有 n 个人去执行 n 个任务，每个人执行每个任务有不同的成功率，每个人只能执行一个任务，求所有任务都执行的总的成功率。

输入第一行，一个整数 n （$1 \le n \le 20$），表示人数兼任务数。接下来 n 行每行 n 个数，第 i 行第 j 个数表示第 i 个人去执行第 j 个任务的成功率（这是一个百分数，在 0 到 100 间）。

[COCI2006-2007 #1] Bond

有 n 个人去执行 n 个任务，每个人执行每个任务有不同的成功率，每个人只能执行一个任务，求所有任务都执行的总的成功率。

输入第一行，一个整数 n （$1 \leq n \leq 20$），表示人数兼任务数。接下来 n 行每行 n 个数，第 i 行第 j 个数表示第 i 个人去执行第 j 个任务的成功率（这是一个百分数，在 0 到 100 间）。

输出最大的总成功率。

[COCI2006-2007 #1] Bond

这算是经典的套路了吧。。

[COCI2006-2007 #1] Bond

这算是经典的套路了吧。。
将乘积转化成log的相加233

[COCI2006-2007 #1] Bond

这算是经典的套路了吧。。

将乘积转化成log的相加233

注意处理0的特殊情况以及精度问题（要用long double）。

DECLARATION

特别声明：作者举的所有例子只是为了更加清晰、通俗地讲述算法流程，可能会有部分触碰机房红线的内容，与本人立场无关。

带花树

匈牙利算法可以用来处理二分图匹配（汉子和妹子匹配），但是如果出现了奇环（Gay），匈牙利算法就无能为力了，那么就要用到带花树。

带花树

匈牙利算法可以用来处理二分图匹配（汉子和妹子匹配），但是如果出现了奇环（Gay），匈牙利算法就无能为力了，那么就要用到带花树。

主要思想是如果发现大小为$2k+1$的班级(环)，那么有一个点不能被匹配，那么就将环中的点先进行男女匹配，剩下的一只单身狗只能去外班找妹子，所以我们可以将一个奇环看做一个点再进行匹配，这个过程叫做开花(blossom)。

带花树的主要过程

将一个汉子进行增广。

带花树的主要过程

将一个汉子进行增广。
增广时的几种情况:

带花树的主要过程

将一个汉子进行增广。

增广时的几种情况:

1. 相邻点是别人的妹子(且本轮已经遍历过)或者和自己同班，直接忽略。

带花树的主要过程

将一个汉子进行增广。
增广时的几种情况:
1. 相邻点是别人的妹子(且本轮已经遍历过)或者和自己同班，直接忽略。
2. 如果是单身，那么视为可以增广，进行匹配并按原路径返回。

带花树的主要过程

将一个汉子进行增广。

增广时的几种情况:

1. 相邻点是别人的妹子(且本轮已经遍历过)或者和自己同班，直接忽略。
2. 如果是单身，那么视为可以增广，进行匹配并按原路径返回。
3. 如果是别人的妹子，那么我们将放到计划列表（队列）中，并看是否可以帮她找到一个新汉子。

带花树的主要过程

将一个汉子进行增广。

增广时的几种情况:

1. 相邻点是别人的妹子(且本轮已经遍历过)或者和自己同班，直接忽略。
2. 如果是单身，那么视为可以增广，进行匹配并按原路径返回。
3. 如果是别人的妹子，那么我们将放到计划列表（队列）中，并看是否可以帮她找到一个新汉子。
4. 对方是个汉子，那么就发现了一个奇环，将整个奇环看做一个班级，全班帮这个汉子一起找妹子。

带花树的主要过程

将一个汉子进行增广。
增广时的几种情况:
1. 相邻点是别人的妹子(且本轮已经遍历过)或者和自己同班，直接忽略。
2. 如果是单身，那么视为可以增广，进行匹配并按原路径返回。
3. 如果是别人的妹子，那么我们将放到计划列表（队列）中，并看是否可以帮她找到一个新汉子。
4. 对方是个汉子，那么就发现了一个奇环，将整个奇环看做一个班级，全班帮这个汉子一起找妹子。
最多增广n次，每次将图遍历一遍，每个点最多开n次花，所以时间复杂度为$O(n^3)$。

带花树的主要过程

将一个汉子进行增广。
增广时的几种情况:
1. 相邻点是别人的妹子(且本轮已经遍历过)或者和自己同班，直接忽略。
2. 如果是单身，那么视为可以增广，进行匹配并按原路径返回。
3. 如果是别人的妹子，那么我们将放到计划列表（队列）中，并看是否可以帮她找到一个新汉子。
4. 对方是个汉子，那么就发现了一个奇环，将整个奇环看做一个班级，全班帮这个汉子一起找妹子。
最多增广n次，每次将图遍历一遍，每个点最多开n次花，所以时间复杂度为$O(n^3)$。
Code

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。
虽然匈牙利不能处理奇环，但是我们可以随机匹配！

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。
虽然匈牙利不能处理奇环，但是我们可以随机匹配！
每次dfs到一个点，先将儿子随机排列，再去增广。

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。

虽然匈牙利不能处理奇环，但是我们可以随机匹配！

每次dfs到一个点，先将儿子随机排列，再去增广。

多跑几遍匈牙利即可，正确率非常高！

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。
虽然匈牙利不能处理奇环，但是我们可以随机匹配！
每次dfs到一个点，先将儿子随机排列，再去增广。
多跑几遍匈牙利即可，正确率非常高！
~~我系渣渣鹏，介系里从未体验过的船新随机算法。~~

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。

虽然匈牙利不能处理奇环，但是我们可以随机匹配！

每次dfs到一个点，先将儿子随机排列，再去增广。

多跑几遍匈牙利即可，正确率非常高！

~~我系渣渣鹏，介系里从未体验过的船新随机算法。~~

~~系兄弟就写随机带花树！！！~~

一种皮皮的随机算法

如果真的学不懂带花树，或者考试的时候调不出来了的话，这里有一种非常皮的随机算法。
虽然匈牙利不能处理奇环，但是我们可以随机匹配！
每次dfs到一个点，先将儿子随机排列，再去增广。
多跑几遍匈牙利即可，正确率非常高！
~~我系渣渣鹏，介系里从未体验过的船新随机算法。~~
~~系兄弟就写随机带花树！！！~~
Code

知乎上的一个题目

$N \times N$的矩阵，取值为0或1，查找最多可以覆盖多少个2×2的、不重叠的正方形（正方形由值为0的格子组成），求好的算法?

知乎上的一个题目

$N \times N$的矩阵，取值为0或1，查找最多可以覆盖多少个2×2的、不重叠的正方形（正方形由值为0的格子组成），求好的算法？

摘自：www.zhihu.com/question/37823835/answer/78633102

知乎上的一个题目

其实就是最小覆盖，只不过是在一般图上。

知乎上的一个题目

其实就是最小覆盖，只不过是在一般图上。
设正方形总数为n，对于所有相交的正方形连边，跑最大匹配的答案为ans，那么答案为$n - ans$。

[WC2016]挑战NPC

有n个球，用整数1到n编号。还有m个筐子，用整数1到m编号。
每个筐子最多能装3个球。
每个球只能放进特定的筐子中。具体有e个条件，第i个条件用两个整数v_i和u_i描述，表示编号为v_i的球可以放进编号为u_i的筐子中。
每个球都必须放进一个筐子中。如果一个筐子内有不超过1个球，那么我们称这样的筐子为半空的。
求半空的筐子最多有多少个，以及在最优方案中，每个球分别放在哪个筐子中。

[WC2016]挑战NPC

非常巧妙的一道题。

[WC2016]挑战NPC

非常巧妙的一道题。
建模方式：考虑将每一个筐子拆成三个点，其中两个点连一条边，每一个球用一个点表示，如果可以放进一个筐子中，那么也连一条边。
如果跑一遍一般图最大匹配的答案为ans，那么答案为$ans - n$。

[WC2016]挑战NPC

非常巧妙的一道题。

建模方式：考虑将每一个筐子拆成三个点，其中两个点连一条边，每一个球用一个点表示，如果可以放进一个筐子中，那么也连一条边。如果跑一遍一般图最大匹配的答案为ans，那么答案为$ans - n$。

原理：球和筐子匹配，表示将该球放进该筐子；筐子和筐子匹配，表示该筐子为半空的。（注意必须先匹配球再匹配筐子，想一想，为什么。）

[WC2016]挑战NPC

非常巧妙的一道题。
建模方式：考虑将每一个筐子拆成三个点，其中两个点连一条边，每一个球用一个点表示，如果可以放进一个筐子中，那么也连一条边。
如果跑一遍一般图最大匹配的答案为ans，那么答案为$ans-n$。
原理：球和筐子匹配，表示将该球放进该筐子；筐子和筐子匹配，表示该筐子为半空的。（注意必须先匹配球再匹配筐子，想一想，为什么。）
Code

[ZOJ3316] Game

给定棋盘上n个旗子,一开始先手可以随便拿,然后每次都不能取离上次的曼哈顿距离超过L的旗子,谁不能动谁输。问后手能否赢?

[ZOJ3316] Game

将距离小于L的点对之间连边，然后跑带花树，如果存在完美匹配，那么后手必胜否则后手必败。

[ZOJ3316] Game

将距离小于L的点对之间连边，然后跑带花树，如果存在完美匹配，那么后手必胜否则后手必败。

因为每次后手可以取先手取的棋子的匹配点。

Thanks

$$Thanks$$