图论选讲

dy0607

雅礼中学

August 14, 2018

Preface

图论是一个几乎每次讲课都有的专题，由于套路的东西实在讲过太多次了，这里会尽量找一些讲得不多的题目。

Preface

图论是一个几乎每次讲课都有的专题，由于套路的东西实在讲过太多次了，这里会尽量找一些讲得不多的题目。

由于是NOIP集训，涉及的算法都是比较简单的常见算法。

Preface

图论是一个几乎每次讲课都有的专题，由于套路的东西实在讲过太多次了，这里会尽量找一些讲得不多的题目。

由于是NOIP集训，涉及的算法都是比较简单的常见算法。

题目难度与顺序无关。

Overview

1. Connectivity
2. Euler Circuit
3. Coloring
4. Shortest Path
5. Spanning Trees
6. Matchings
7. Flows

复习一些定义

强连通分量：在有向图G中，如果u能到v，v能到u，则称u和v强连通。如果一个图中任意两点强连通，则称该图为强连通图。有向图的极大强连通子图称为强连通分量。有向图缩强连通分量后是一个DAG（有向无环图）。

复习一些定义

强连通分量：在有向图G中，如果u能到v，v能到u，则称u和v强连通。如果一个图中任意两点强连通，则称该图为强连通图。有向图的极大强连通子图称为强连通分量。有向图缩强连通分量后是一个DAG（有向无环图）。

点(边)双连通分量：在无向图G中，任意删去一个点（一条边），如果连通性保持不变，则称该图点（边）双联通。无向图的极大点（边）双联通子图称为点（边）双联通分量。

以上均可以用Tarjan在$O(n+m)$内解决。

复习一些定义

2-satisfiability：n个布尔变量，有若干形如"x_i为真/假不能与x_j为真/假同时成立"的限制，要求给出一组解。方法是将每个变量拆成真/假两个点，然后将限制连边求强连通分量，判断两个点是否在一个分量内。

复习一些定义

2-satisfiability：n个布尔变量，有若干形如“x_i为真/假不能与x_j为真/假同时成立”的限制，要求给出一组解。方法是将每个变量拆成真/假两个点，然后将限制连边求强连通分量，判断两个点是否在一个分量内。

传递闭包：对于一个有向图G，求出从每一个点出发可以到达哪些点。方法是先缩强连通分量，然后在DAG上dp。

传递闭包问题的常用技巧是bitset优化获得一个$\frac{1}{64}$的常数。

SDOI2010 所驼门王的宝藏

有一个$r \times c$的网格，网格的n个格子放着宝藏，每个有宝藏的格子都有一个传送门，传送门是以下三种中的一种：

- “横天门”：由该门可以传送到同行的任一格子；
- “纵寰门”：由该门可以传送到同列的任一格子；
- “自由门”：由该门可以传送到以该门所在格子为中心周围8格中任一格子。

找一条路径（起点终点任意，可以经过重复的点）经过尽量多的宝藏（重复经过只算一次）。

$n \le 10^5, r, c \le 10^6$

SDOI2010 所驼门王的宝藏

从传送门直接向可以传送到的有宝藏的格子连边，然后缩强连通分量，得到一个DAG.

SDOI2010 所驼门王的宝藏

从传送门直接向可以传送到的有宝藏的格子连边，然后缩强连通分量，得到一个DAG.

答案就是DAG上$size$之和最大的链，可以简单地dp得到答案。但这样边数是$O(n^2)$的。

SDOI2010 所驼门王的宝藏

从传送门直接向可以传送到的有宝藏的格子连边，然后缩强连通分量，得到一个DAG.

答案就是DAG上$size$之和最大的链，可以简单地dp得到答案。但这样边数是$O(n^2)$的。

对于所有行和列额外建一个点，向这行/列的每一个有宝藏的格子连边，然后有传送整行/列的传送门就可以直接向新建的点连边了.

$O(r+c+n)$

HNOI2012 矿场搭建

煤矿工地可以看成是由隧道连接挖煤点组成的无向图。为安全起见，希望在工地发生事故时所有挖煤点的工人都能有一条出路逃到救援出口处。于是矿主决定在某些挖煤点设立救援出口，使得无论哪一个挖煤点坍塌之后，其他挖煤点的工人都有一条道路通向救援出口。

计算至少需要设置几个救援出口，以及不同最少救援出口的设置方案总数。

$m \leq 500$

HNOI2012 矿场搭建

找出所有点双连通分量以及割点.

HNOI2012 矿场搭建

找出所有点双连通分量以及割点.

如果一个双联通分量中只有一个割点，那么必须在这个双联通分量的非割点位置放一个出口.

如果整个连通块是一个双连通分量（也就是没有割点），那么必须任意放两个出口.

HNOI2012 矿场搭建

找出所有点双连通分量以及割点.

如果一个双联通分量中只有一个割点，那么必须在这个双联通分量的非割点位置放一个出口.

如果整个连通块是一个双连通分量（也就是没有割点），那么必须任意放两个出口.

可以发现这样放就够了, $O(n)$.

51NOD1318 最大公约数与最小公倍数方程组

求解一个有趣的方程组，方程组有n个未知正整数x_i。

方程组由M个方程组成，方程只有两种类型：

- $gcd(x_i, x_j) = p$
- $lcm(x_i, x_j) = q$

你需要判断这样一个方程组是否存在正整数解。

$T \leq 10, n \leq 200, p, q \leq 10^9$

51NOD1318 最大公约数与最小公倍数方程组

对每个方程中右边的数质因数分解，设$cnt(i,p)$为x_i中有多少个质因子p，那么我们实际上得到的是这样一些限制：

- $\max(cnt(i,p), cnt(j,p)) = a$
- $\min(cnt(i,p), cnt(j,p)) = b$

这个问题可以用2-sat解决：变量$a[i][p][j]$表示是否有$p^j \mid x_i$，上面的限制就可以很容易地表示出来。

当然如果某个变量的某个质因子在方程组中没有涉及，就不需要建出点了，这样复杂度为$O(Tm\log 10^9)$

复习一些定义

欧拉回路：一条路径经过所有的边恰好一次，最后回到起点。欧拉路径的区别是不要求回到起点。

复习一些定义

欧拉回路：一条路径经过所有的边恰好一次，最后回到起点。欧拉路径的区别是不要求回到起点。

存在欧拉回路的充要条件是所有点度数为偶数，且除去孤立点后图连通。欧拉路径则允许有两个点度数为奇数。

欧拉回路可以用DFS在线性时间内求出。

CF 296 div1 C

n个点m条边的连通无向图，加尽量少的边（可以有重边自环），使得存在一种给边定向的方案，方案中每个点的入度和出度都是偶数。输出方案。

$n \leq 10^5, m \leq 2 \times 10^5$

CF 296 div1 C

这类问题有一个比较常见的思路：找一个答案的下界，然后尝试通过构造达到这个下界。

CF 296 div1 C

这类问题有一个比较常见的思路：找一个答案的下界，然后尝试通过构造达到这个下界。

首先每个点的度数一定是偶数，我们可以先把度数是奇数的点两两连起来。然后边的个数一定是偶数，如果此时边的个数是奇数，那么还要加上一条边。这显然是一个答案的下界。

CF 296 div1 C

这类问题有一个比较常见的思路：找一个答案的下界，然后尝试通过构造达到这个下界。

首先每个点的度数一定是偶数，我们可以先把度数是奇数的点两两连起来。然后边的个数一定是偶数，如果此时边的个数是奇数，那么还要加上一条边。这显然是一个答案的下界。

如果边的个数是偶数考虑怎么构造答案，注意到此时图中一定存在欧拉回路，如果存在一条回路$a-b-c-d-a$，我们我们可以这样给边交错地定向：$a > b < c > d < a$，这样经过的每个点要么入度增加2，要么出度增加2，一定会满足条件。

CF 296 div1 C

这类问题有一个比较常见的思路：找一个答案的下界，然后尝试通过构造达到这个下界。

首先每个点的度数一定是偶数，我们可以先把度数是奇数的点两两连起来。然后边的个数一定是偶数，如果此时边的个数是奇数，那么还要加上一条边。这显然是一个答案的下界。

如果边的个数是偶数考虑怎么构造答案，注意到此时图中一定存在欧拉回路，如果存在一条回路$a-b-c-d-a$，我们我们可以这样给边交错地定向：$a > b < c > d < a$，这样经过的每个点要么入度增加2，要么出度增加2，一定会满足条件。

边的个数是奇数时加上一个自环即可，$O(n+m)$.

CF 245 div1 E

一个全为0的序列，有n段区间$[l_i, r_i]$，你需要给每一段区间$+1$或者-1，使得操作结束后序列中的所有位置绝对值不超过1。输出方案。

$n \le 10^5, 1 \le l_i \le r_i \le 10^9$

CF 245 div1 E

考虑对序列差分，那么区间加$[l,r]$就相当于在差分的序列上将l增加1，将$r+1$减去1. 区间减是类似的.

CF 245 div1 E

考虑对序列差分，那么区间加$[l,r]$就相当于在差分的序列上将l增加1，将$r+1$减去1. 区间减是类似的.

发挥想象力，发现这很像有一条无向边$(l,r+1)$，加减两种决策就是给这条边定向，每个点的值就是入度与出度的差。

CF 245 div1 E

考虑对序列差分，那么区间加$[l,r]$就相当于在差分的序列上将l增加1，将$r+1$减去1. 区间减是类似的.

发挥想象力，发现这很像有一条无向边$(l,r+1)$，加减两种决策就是给这条边定向，每个点的值就是入度与出度的差。

先分析简单的情况：如果每个点的度数都是偶数，那么对于每个连通块求欧拉回路，这样每个点的入度等于出度，也就是差分序列全为0，原序列也是全为0.

对于度数是奇数的点，仍然采用上一题的做法，将奇数点从左往右依次两两配对连边，再求欧拉回路即可。

$O(n\log n)$

复习一些定义

独立集：一个点集中两两之间没有连边，则称这个点集是一个独立集。

复习一些定义

独立集：一个点集中两两之间没有连边，则称这个点集是一个独立集。

团: 一个点集中两两之间都有连边，则称这个点集是一个团。最大团可以在$O(3^{\frac{n}{3}})$内求解。

复习一些定义

独立集：一个点集中两两之间没有连边，则称这个点集是一个独立集。

团：一个点集中两两之间都有连边，则称这个点集是一个团。最大团可以在$O(3^{\frac{n}{3}})$内求解。

图染色问题：判断一个图能否被划分为k个不相交的独立集。$k > 2$时为NP问题。

HNOI2010 平面图判定

给出一个存在哈密顿回路的无向图以及它的哈密顿回路，问这张图是否是平面图。

$T \leq 100, n \leq 200, m \leq 10^4$，没有重边自环

HNOI2010 平面图判定

哈密顿回路构成了一个圈，考虑除哈密顿回路以外的边，我们可以从圈内连接这两个点，也可以从圈外连接。

HNOI2010 平面图判定

哈密顿回路构成了一个圈，考虑除哈密顿回路以外的边，我们可以从圈内连接这两个点，也可以从圈外连接。

有些边是不能放在同一边的，我们将冲突的边两两连边。我们只需要判断连出来的图是不是二分图。

HNOI2010 平面图判定

哈密顿回路构成了一个圈，考虑除哈密顿回路以外的边，我们可以从圈内连接这两个点，也可以从圈外连接。

有些边是不能放在同一边的，我们将冲突的边两两连边。我们只需要判断连出来的图是不是二分图。

注意到$m > 3 \times n$时肯定不合法，可以直接跳过。这样复杂度为$O(Tn^2)$。

Goodbye2017 H

一个n个点的有向图，设$f(u,v)$表示是否存在一条u到v的路径。对于每一对$1 \le u < v \le n$，你知道它们至少满足下面的某个条件（输入会对每一对u,v给出这个条件）：

- $f(u,v)$ and $f(v,u) = true$
- $f(u,v)$ xor $f(v,u) = true$
- $f(u,v)$ or $f(v,u) = true$

求这张图至少有多少条边，或者判断无解。

$n \le 47.$

Goodbye2017 H

如果满足and为真，那么两个点一定在一个强连通分量内；如果xor为真则一定不在一个中。我们将一定在一个强连通分量内的点缩点，并判掉无解的情况。

Goodbye2017 H

如果满足and为真，那么两个点一定在一个强连通分量内；如果xor为真则一定不在一个中。我们将一定在一个强连通分量内的点缩点，并判掉无解的情况。

由于三个条件中一定会满足一个，而我们又要最小化边的数量，这个图最后一定会形如一些强连通分量连成一条链。构造大于一个点的强连通分量，最省边的方法是连成一个环，因此答案就是$(n-1+$大小大于1的强连通分量个数$)$。

Goodbye2017 H

如果满足and为真，那么两个点一定在一个强连通分量内；如果xor为真则一定不在一个中。我们将一定在一个强连通分量内的点缩点，并判掉无解的情况。

由于三个条件中一定会满足一个，而我们又要最小化边的数量，这个图最后一定会形如一些强连通分量连成一条链。构造大于一个点的强连通分量，最省边的方法是连成一个环，因此答案就是$(n-1+$ 大小大于1的强连通分量个数 $)$。

于是对于大小大于1的强连通分量（最多$\frac{n}{2}$个），我们希望将它们合并成尽量少的块，而两个强连通分量可以合并当且仅当它们之间没有xor的关系，我们实际要算一个图的最小染色数。

Goodbye2017 H

判断一张任意图能否被k染色可以在$O(2^n)$内解决。考虑一个看起来更难的问题：有多少种选择k个独立集的方案，使得它们覆盖了整张图，（独立集之间可以有交集）。显然一张图可以被k染色当且仅当这个问题的答案大于0。

判断一张任意图能否被k染色可以在$O(2^n)$内解决。考虑一个看起来更难的问题：有多少种选择k个独立集的方案，使得它们覆盖了整张图，（独立集之间可以有交集）。显然一张图可以被k染色当且仅当这个问题的答案大于0。

这个问题可以用容斥解决，大概就是用任意选择k个的方案减去未完全覆盖的方案：

$$ans = \sum_{S \in V} (-1)^{n-|S|} cnt^k(S)$$

其中$cnt(S)$表示S内有多少独立集.

Goodbye2017 H

$cnt(S)$可以用简单的状压在$O(2^n)$内得到(考虑独立集是否包含某个点u)：

$$cnt(S) = cnt(S-u) + cnt(S-u-adj(u))$$

$adj(u)$表示与u相邻的点的集合。

Goodbye2017 H

$cnt(S)$可以用简单的状压在$O(2^n)$内得到(考虑独立集是否包含某个点u)：

$$cnt(S) = cnt(S-u) + cnt(S-u-adj(u))$$

$adj(u)$表示与u相邻的点的集合。

由于方案数可能会比较多，我们需要在模大质数意义下算，这样错误率可以接受。$O(n2^{\frac{n}{2}})$

复习一些定义

最短路：对于带权无向图G，最短路是一条S到T权值和最小的路径。

复习一些定义

最短路：对于带权无向图G，最短路是一条S到T权值和最小的路径。

Floyd：$O(n^3)$算出所有点对之间的最短路。

复习一些定义

最短路：对于带权无向图G，最短路是一条S到T权值和最小的路径。

Floyd：$O(n^3)$算出所有点对之间的最短路。

Dijkstra: 暴力$O(n^2)$，一般的堆是$O((n+m)\log n)$，斐波那契堆可以做到$O(m+n\log n)$.

复习一些定义

最短路：对于带权无向图G，最短路是一条S到T权值和最小的路径。

Floyd：$O(n^3)$算出所有点对之间的最短路。

Dijkstra: 暴力$O(n^2)$，一般的堆是$O((n+m)\log n)$, 斐波那契堆可以做到$O(m+n\log n)$.

Bellman-Ford / SPFA：$O(m)$或者$O(nm)$，取决于出题人。

复习一些定义

最短路：对于带权无向图G，最短路是一条S到T权值和最小的路径。

Floyd：$O(n^3)$算出所有点对之间的最短路。

Dijkstra: 暴力$O(n^2)$，一般的堆是$O((n+m)\log n)$, 斐波那契堆可以做到$O(m+n\log n)$.

Bellman-Ford / SPFA：$O(m)$或者$O(nm)$，取决于出题人。

SPFA的特殊用途是用来找负权环。

APIO2017 商旅

n个城市，m条单向道路，每条道路有一个长度，k种商品。每个城市都可以卖出或购买某些商品，每个城市卖出获得的收益一定小于等于购买需要的花费，各个城市之间的价格可能不同，因此你可以通过赚取差价来获得收益。

你最多只能持有一个商品，你需要找一个环（可以多次经过一个点或边），最大化走一遍环的收益与环长的比值。

$n \le 100, m \le 9900, k \le 1000$

APIO2017 商旅

首先求出最短路，对每个城市对(u, v)，求出从u市购买到v市卖出能获得的最大收益w。建一个新图，u到v的距离dis为原来的最短路的长度，边权为w。那么我们要求一个环，边权与环长比值最大。

APIO2017 商旅

首先求出最短路，对每个城市对(u,v)，求出从u市购买到v市卖出能获得的最大收益w。建一个新图，u到v的距离dis为原来的最短路的长度，边权为w。那么我们要求一个环，边权与环长比值最大。

二分答案k，我们要判断是否有$\frac{\sum w}{\sum dis} \geq k$，
即$\sum(w - k \times dis) \geq 0$。将边权变为$k \times dis - w$，判断是否存在非正权环即可。

$O(n^2k + nm)$

NOI2010 海拔

一个$(n+1)\times(n+1)$的点阵，每个点有一个海拔，你只知道左上角海拔为0，右下角海拔为1。四连通的点之间有双向道路连接，走过一条边耗费的体力是$\max(0,$ 终点海拔 $-$ 起点海拔$)$。每条边的每个方向都有一定数量的行人走过，你需要计算所有行人耗费的体力之和至少是多少。

$80\% : n \leq 40$

$100\% : n \leq 500$

NOI2010 海拔

首先发现最优情况下海拔一定为0/1，并且存在这样一条割，使得割的两部分海拔分别为0/1.

NOI2010 海拔

首先发现最优情况下海拔一定为0/1，并且存在这样一条割，使得割的两部分海拔分别为0/1.

将经过边的行人作为流量，我们实际上要求一个最小割。直接做可以拿到80分。

NOI2010 海拔

首先发现最优情况下海拔一定为0/1，并且存在这样一条割，使得割的两部分海拔分别为0/1.

将经过边的行人作为流量，我们实际上要求一个最小割。直接做可以拿到80分。

注意到这是个平面图，而平面图上的最小割可以转化为其对偶图上的最短路，$O(n^2 \log n)$.

USACO07FEB Lilipad Pond

一个$n \times m$的池塘，一些位置上放置了荷叶叶。给定起点，你需要通过走马步（坐标某一维变化1，另一维变化2）到达终点，你不能跳到水里，每一步都必须在荷叶上(起点和终点都在荷叶上)。你需要算出最少还需要放置多少荷叶叶才能到达终点，以及有多少种最少放置的放置方法。

$n, m \le 30$

USACO07FEB Lilipad Pond

第一问非常简单，连边时如果边的终点没有荷叶，则边权为1，否则为0，跑最短路即可。

USACO07FEB Lilipad Pond

第一问非常简单，连边时如果边的终点没有荷叶，则边权为1，否则为0，跑最短路即可。

然而在这张图上算出的最短路个数并不是第二问的答案，我们需要改变一下最短路建边方式。我们只对起点和没有荷叶的点建点，对于每个点，考虑它走到的下一个没有荷叶的点可能是哪些，连边权为1的边。再求最短路方案数即可。

$O(n^2m^2)$.

AIM Tech 4 div1 D

一张带权无向图，q次操作：

- 1 v 询问1到v的最短路
- 2 c l_1 l_2 ... l_c 将边l_i的权值增加1

$n, m \leq 10^5, q \leq 2000, \sum c \leq 10^6, TL = 10s.$

AIM Tech 4 div1 D

Dijkstra有个这样的技巧：如果所有点的距离都小于某个较小的值，我们就可以不用堆，而是开值域个队列，这样最短路就变为线性了。

AIM Tech 4 div1 D

Dijkstra有个这样的技巧：如果所有点的距离都小于某个较小的值，我们就可以不用堆，而是开值域个队列，这样最短路就变为线性了。

先求一遍最短路，之后每次修改后重新算最短路时，每个点距离的增加量是$\leq c$的。这样我们在Dijkstra时，每次找出距离的增加量最小的点进行松弛，可以发现这样仍是对的。而由于距离增加量的值域很小，就可以用队列代替堆解决。

$O((n+m)q+\sum c)$

复习一些定义

生成树：无向图的一个连通子图，恰好有$n-1$条边。最小生成树主要有Prim和Kruskal算法。

复习一些定义

生成树：无向图的一个连通子图，恰好有$n-1$条边。最小生成树主要有Prim和Kruskal算法。

Prim：$O((n+m)\log n)$或用斐波那契堆做到$O(m+n\log n)$。

Kruskal：$O(m\log m)$，可配合基尔霍夫矩阵进行最小生成树计数。

复习一些定义

生成树：无向图的一个连通子图，恰好有$n-1$条边。最小生成树主要有Prim和Kruskal算法。

Prim：$O((n+m)\log n)$或用斐波那契堆做到$O(m+n\log n)$。

Kruskal：$O(m\log m)$，可配合基尔霍夫矩阵进行最小生成树计数。

树形图：有向图的一个子图，从某个点可以到达其它所有点。最小树形图一般用朱刘算法。

复习一些定义

生成树：无向图的一个连通子图，恰好有$n-1$条边。最小生成树主要有Prim和Kruskal算法。

Prim：$O((n+m)\log n)$或用斐波那契堆做到$O(m+n\log n)$。

Kruskal：$O(m\log m)$，可配合基尔霍夫矩阵进行最小生成树计数。

树形图：有向图的一个子图，从某个点可以到达其它所有点。最小树形图一般用朱刘算法。

朱刘算法：朴素实现为$O(nm)$，后被Tarjan优化到$O(m\log n)$.

CF 434 div1 D

一个无向图，每次可以删去两条有公共点的边，求最多能删去多少条边，输出方案。

$n, m \leq 2 \times 10^5$，无重边自环

CF 434 div1 D

题目可以这样理解：将每条边放在两个端点中的一个上，使得尽量多的点上的边的数量为偶数。

CF 434 div1 D

题目可以这样理解：将每条边放在两个端点中的一个上，使得尽量多的点上的边的数量为偶数。

每个连通块分开考虑，随意求出一棵生成树，对于非树边任意分配。然后考虑分配树边，对于一个点处理完它子树后，如果它的边的数量为奇数，则将它到它父亲的边分配给子集，否则给父亲。这样最多只有根无法变为偶数，已经达到了答案上界。

$O(n+m)$

SCOI2012 滑雪

n个点m条边的带权图，每个点有一个高度。从一号点开始，每次选择一种操作：

- 瞬移到曾经到过的某个点
- 选择一条边走过去，要求终点的高度不高于起点的高度

求最多能经过多少点，在经过尽量多的点的前提下最小化走过的边权之和。

$n \leq 10^5, m \leq 10^6$

SCOI2012 滑雪

最后得到的路径是一棵树，但直接最小生成树又可能无法满足高度的限制。

SCOI2012 滑雪

最后得到的路径是一棵树，但直接最小生成树又可能无法满足高度的限制。

在做Prim时，以高度为第一关键字，距离为第二关键字，优先拓展高度高的节点。这样一个节点一定是由高度较高的节点拓展而来的，因此保证了合法性.

$O(m \log m)$

复习一些定义

匹配：对于无向图G，一组匹配是一个边集，所有点至多是其中一条边的端点。完美匹配则要求所有点都在匹配中。

复习一些定义

匹配：对于无向图G，一组匹配是一个边集，所有点至多是其中一条边的端点。完美匹配则要求所有点都在匹配中。

二分图匹配有$O(nm)$的匈牙利算法以及$O(\sqrt{n}m)$的Hopcraft-Karp算法（其实就是Dinic）。

复习一些定义

匹配：对于无向图G，一组匹配是一个边集，所有点至多是其中一条边的端点。完美匹配则要求所有点都在匹配中。

二分图匹配有$O(nm)$的匈牙利算法以及$O(\sqrt{n}m)$的Hopcraft-Karp算法（其实就是Dinic）。

霍尔定理给出了一个存在完美匹配的充要条件：对于同一边的任意点集S，与S相邻的点的个数大于等于S的点数。

复习一些定义

匹配：对于无向图G，一组匹配是一个边集，所有点至多是其中一条边的端点。完美匹配则要求所有点都在匹配中。

二分图匹配有$O(nm)$的匈牙利算法以及$O(\sqrt{n}m)$的Hopcraft-Karp算法（其实就是Dinic）。

霍尔定理给出了一个存在完美匹配的充要条件：对于同一边的任意点集S，与S相邻的点的个数大于等于S的点数。

二分图最大权匹配存在$O(n^3)$的KM算法，也可以用费用流做.

复习一些定义

匹配：对于无向图G，一组匹配是一个边集，所有点至多是其中一条边的端点。完美匹配则要求所有点都在匹配中。

二分图匹配有$O(nm)$的匈牙利算法以及$O(\sqrt{n}m)$的Hopcraft-Karp算法（其实就是Dinic）。

霍尔定理给出了一个存在完美匹配的充要条件：对于同一边的任意点集S，与S相邻的点的个数大于等于S的点数。

二分图最大权匹配存在$O(n^3)$的KM算法，也可以用费用流做.

一般图最大匹配需用带花树，$O(n^2m)$，后被优化到$O(\sqrt{n}m)$.

复习一些定义

最大独立集：等于$n-$最大匹配数。

复习一些定义

最大独立集：等于$n-$最大匹配数。

最小边覆盖：选中尽量少的边，使得所有点至少是一条选中的边的端点。等于$n-$最大匹配数。

复习一些定义

最大独立集：等于$n-$最大匹配数。

最小边覆盖：选中尽量少的边，使得所有点至少是一条选中的边的端点。等于$n-$最大匹配数。

最小点覆盖：选中尽量少的点，使得所有边有至少一个端点被选中。等于最大匹配数。

ZJOI2007 矩阵游戏

一个$n \times n$的网格，每个格子有黑白两种颜色。你可以交换两行或交换两列（可以进行任意次操作），使得主对角线(左上到右下)上的格子均为黑色。

判断是否有解，$n \leq 200$。

ZJOI2007 矩阵游戏

我们实际上要找是否存在一个排列P，$\forall i \in [1, n], (i, P_i)$为黑色。
如果存在我们就可以通过行交换造出主对角线，而列交换并不会对是否存在产生影响。

ZJOI2007 矩阵游戏

我们实际上要找是否存在一个排列P，$\forall i \in [1, n]$, (i, P_i)为黑色。如果存在我们就可以通过行交换造出主对角线，而列交换并不会对是否存在产生影响。

对行和列分别建n个点，如果第i行第j列为黑色，则将第i行和第j列连边。这样我们得到一个二分图，原问题有解当且仅当这个二分图存在完美匹配.

ZJOI2007 矩阵游戏

我们实际上要找是否存在一个排列P，$\forall i \in [1, n]$, (i, P_i)为黑色。如果存在我们就可以通过行交换造出主对角线，而列交换并不会对是否存在产生影响。

对行和列分别建n个点，如果第i行第j列为黑色，则将第i行和第j列连边。这样我们得到一个二分图，原问题有解当且仅当这个二分图存在完美匹配.

$O(n^3)$

NOI2009 变换序列

有一个n的排列P，定义$D(x, y) = \min(|x - y|, n - |x - y|)$，给出$D(i, P_i)$，求出满足条件的字典序最小的$P$，或者判断无解。

$n \le 10^4$.

NOI2009 变换序列

对于每个i, 备选的P_i最多只有两个。我们还需要使得字典序最小，一种暴力方法是从前往后逐位确定，先将P_i设为较小的那个候选值，再判断在前面的匹配不变的情况下后面的点是否能够完美匹配，不能则设为较大的，这样是$O(n^3)$的。

NOI2009 变换序列

对于每个i，备选的P_i最多只有两个。我们还需要使得字典序最小，一种暴力方法是从前往后逐位确定，先将P_i设为较小的那个候选值，再判断在前面的匹配不变的情况下后面的点是否能够完美匹配，不能则设为较大的，这样是$O(n^3)$的。

另一种方法是从后往前做匈牙利算法，每次贪心地选最小的点尝试进行匹配（也就是把边表从小到大排序），这样是$O(n^2)$的。注意它用到了备选点不超过2的性质。

NOI2009 变换序列

对于每个i，备选的P_i最多只有两个。我们还需要使得字典序最小，一种暴力方法是从前往后逐位确定，先将P_i设为较小的那个候选值，再判断在前面的匹配不变的情况下后面的点是否能够完美匹配，不能则设为较大的，这样是$O(n^3)$的。

另一种方法是从后往前做匈牙利算法，每次贪心地选最小的点尝试进行匹配（也就是把边表从小到大排序），这样是$O(n^2)$的。注意它用到了备选点不超过2的性质。

另一种做法是，如果备选点为u, v，那么我们可以连一条u到v的边，显然只有每个连通块的点数等于边数时才有解，也就是每个连通块都是环套树。对于树边可以直接确定，环上也只有两种决策，可以做到线性。

HAOI2017 新型城市化

有一个n个点的无向图，其中只有m对点之间没有连边，保证这张图可以被分为至多两个团.

对于m对未连边的点对，判断有哪些点对满足将他们连边后最大团的大小增加.

$n \leq 10^4, m \leq 1.5 \times 10^5.$

HAOI2017 新型城市化

考虑将原图取反，也就是输入的m条边构成的图．这个图一定是个二分图，因为题目保证了原图可以被至多两个团覆盖．

HAOI2017 新型城市化

考虑将原图取反，也就是输入的m条边构成的图．这个图一定是个二分图，因为题目保证了原图可以被至多两个团覆盖．

原图上的最大团，也就是反图上的最大独立集，而二分图的最大独立集等于点数减去最大匹配数．那么题目就是问去掉哪些边后最大匹配数减少，也就是哪些边一定在最大匹配上．

HAOI2017 新型城市化

考虑将原图取反，也就是输入的m条边构成的图．这个图一定是个二分图，因为题目保证了原图可以被至多两个团覆盖．

原图上的最大团，也就是反图上的最大独立集，而二分图的最大独立集等于点数减去最大匹配数．那么题目就是问去掉哪些边后最大匹配数减少，也就是哪些边一定在最大匹配上．

这题中n, m较大，需要用Dinic算二分匹配．怎么判断哪些边一定在最大匹配上?

HAOI2017 新型城市化

考虑将原图取反，也就是输入的m条边构成的图．这个图一定是个二分图，因为题目保证了原图可以被至多两个团覆盖．

原图上的最大团，也就是反图上的最大独立集，而二分图的最大独立集等于点数减去最大匹配数．那么题目就是问去掉哪些边后最大匹配数减少，也就是哪些边一定在最大匹配上．

这题中n, m较大，需要用Dinic算二分匹配．怎么判断哪些边一定在最大匹配上？

显然它们一定要满流，其次边上的两个点在残量网络上不能在同一个强连通分量中，因为如果他们在同一个环中，就可以将环上未匹配的边设为匹配边，匹配边设为未匹配边，最大匹配不变．$O(m\sqrt{n})$

复习一些定义

最小割：割掉总权值尽量少的边，使得S，T不连通。

最大流最小割定理：S到T的最大流等于S到T的最小割。这是线性规划中对偶原理的一种特殊情况。

残量网络：未满流的边构成的网络。

复习一些定义

最小割：割掉总权值尽量少的边，使得S，T不连通。

最大流最小割定理：S到T的最大流等于S到T的最小割。这是线性规划中对偶原理的一种特殊情况。

残量网络：未满流的边构成的网络。

Edmond-Karp复杂度为$O(nm^2)$；Dinic复杂度为$O(n^2m)$，~~写个动态树可以优化到$O(nm\log n)$~~，但在特殊图以及随机图上有更好的复杂度。

wiki上关于网络流最优的复杂度为$O(nm)$.

AHOI2009 最小割

给出一个有向图和源点汇点，求哪些边可能在最小割上，哪些边一定在最小割上。

$n \leq 4 \times 10^3, m \leq 6 \times 10^4$

AHOI2009 最小割

是否可能：边(u, v)满流，且残量网络上u到v不存在路径。
即u和v不在同一强连通分量内。

AHOI2009 最小割

是否可能：边(u,v)满流，且残量网络上u到v不存在路径。
即u和v不在同一强连通分量内。

是否一定: 边(u,v)满流，且残量网络上存在S到u的路径，v到T的路径。即S和u在同一强连通分量内，v和T在同一强连通分量内。

SDOI2014 LIS

给定序列A，序列中的每一项A_i有删除代价B_i和附加属性C_i。删除若干项，使得A的最长上升子序列长度减少至少1，且付出的代价之和最小，并输出方案。如果有多种方案，请输出将删去项的附加属性排序之后，字典序最小的一种。

$n \leq 700$

SDOI2014 LIS

先用$O(n^2)$的dp做最长上升子序列，如果$i < j, dp[j] = dp[i] + 1$，则由i向j连边。这样我们得到一个DAG，我们需要删掉权值和最小的点，使得DAG中最长链的长度减少1。

SDOI2014 LIS

先用$O(n^2)$的dp做最长上升子序列，如果$i < j, dp[j] = dp[i] + 1$，则由i向j连边。这样我们得到一个DAG，我们需要删掉权值和最小的点，使得DAG中最长链的长度减少1。

将每个点拆成两个点，将所有入边连到第一个点，第二个点连所有出边，中间连流量为权值的边，代表割掉这个点的代价。源点向$dp[i] = 1$的位置连∞的边，$dp[i] = Max$的位置向汇点连∞的边。这样求最小割就就行了，但我们还需要考虑字典序最小。

SDOI2014 LIS

先用$O(n^2)$的dp做最长上升子序列，如果$i < j, dp[j] = dp[i] + 1$，则由i向j连边。这样我们得到一个DAG，我们需要删掉权值和最小的点，使得DAG中最长链的长度减少1。

将每个点拆成两个点，将所有入边连到第一个点，第二个点连所有出边，中间连流量为权值的边，代表割掉这个点的代价。源点向$dp[i] = 1$的位置连∞的边，$dp[i] = Max$的位置向汇点连∞的边。这样求最小割就就行了，但我们还需要考虑字典序最小。

一种方法是按C_i从小到大确定是否可能在割上，如果可能，则将它加入答案，并且将这个点删去。但此时残量网络会发生改变，而如果重新做最小割又太慢了.

SDOI2014 LIS

先用$O(n^2)$的dp做最长上升子序列，如果$i < j, dp[j] = dp[i] + 1$，则由i向j连边。这样我们得到一个DAG，我们需要删掉权值和最小的点，使得DAG中最长链的长度减少1。

将每个点拆成两个点，将所有入边连到第一个点，第二个点连所有出边，中间连流量为权值的边，代表割掉这个点的代价。源点向$dp[i] = 1$的位置连∞的边，$dp[i] = Max$的位置向汇点连∞的边。这样求最小割就就行了，但我们还需要考虑字典序最小。

一种方法是按C_i从小到大确定是否可能在割上，如果可能，则将它加入答案，并且将这个点删去。但此时残量网络会发生改变，而如果重新做最小割又太慢了.

于是有一种退流的技巧：从这个点向源点和汇点分别做最小割，就能将原来流到这个点的流量退回去。

CF Edu 43 F

对于一个二分图，定义最小k-点覆盖为一个最小的边集，使得所有点被覆盖至少k次。设$Mindeg$为所有点度数的最小值，对所有$k \in [0, Mindeg]$，求最小k点覆盖的大小，并输出方案。

$n, m \leq 2 \times 10^3$

CF Edu 43 F

回忆一下正常的最小点覆盖是怎么做的，它等于$n-$最大匹配数。

CF Edu 43 F

回忆一下正常的最小点覆盖是怎么做的，它等于$n-$最大匹配数。

这里我们仍然用最大匹配的建边，只不过源点到每个点的边以及每个点到汇点的边的流量上限变为k。答案是$nk-$最大流。

$O((n+m)^2)$

CF Edu 31 F

我们称一个串是“反回文”的，当且仅当$\forall i \in [1,n], S_i \neq S_{n+1-i}$.

给出一个串s，你需要找到它的一个排列t，满足t是反回文的。给出w_i，定义t的优美度为$\sum_{i=1}^{n}[s_i = t_i]w_i$。输出满足条件的情况下优美度的最大值。

串中仅包含小写字母，$n \leq 100$，n为偶数，保证有解。

CF Edu 31 F

对于每个字母建一个点，由源点向它连流量为出现次数的边，费用为0; 然后对于每对对称的位置建一个点，由它向汇点连流量为2的边，费用为0.

CF Edu 31 F

对于每个字母建一个点，由源点向它连流量为出现次数的边，费用为0; 然后对于每对对称的位置建一个点，由它向汇点连流量为2的边，费用为0.

每个字母c到每对位置$(i, n+1-i)$之间连边，流量为1（这里保证了串是反回文的），费用需要进行讨论:

CF Edu 31 F

对于每个字母建一个点，由源点向它连流量为出现次数的边，费用为0; 然后对于每对对称的位置建一个点，由它向汇点连流量为2的边，费用为0.

每个字母c到每对位置$(i, n+1-i)$之间连边，流量为1（这里保证了串是反回文的），费用需要进行讨论:

- 若$s_i = s_{n+1-i} = c$，费用为$-\max\{w_i, w_{n+1-i}\}$.
- 若$s_i = c$，费用为$-w_i$；若$s_{n+1-i} = c$，费用为$-w_{n+1-i}$.
- 否则费用为0.

CF Edu 31 F

对于每个字母建一个点，由源点向它连流量为出现次数的边，费用为0; 然后对于每对对称的位置建一个点，由它向汇点连流量为2的边，费用为0.

每个字母c到每对位置$(i, n+1-i)$之间连边，流量为1（这里保证了串是反回文的），费用需要进行讨论:

- 若$s_i = s_{n+1-i} = c$，费用为$-\max\{w_i, w_{n+1-i}\}$.
- 若$s_i = c$，费用为$-w_i$；若$s_{n+1-i} = c$，费用为$-w_{n+1-i}$.
- 否则费用为0.

求最小费用最大流即可，对于中间的边费用都加上一个较大的值K避免负权，最后答案为$nK-$最小费用。

NOI2012 美食节

共有n种不同的菜品，有p_i个同学点了第i种菜品，有m个厨师来制作这些菜品。当所有的同学点餐结束后，菜品的制作任务就会分配给每个厨师。然后每个厨师就会同时开始做菜。厨师们会按照要求的顺序进行制作，并且每次只能制作一人份。第i个厨师制作第j种菜品的时间记为$t[i][j]$。

如果一个同学点的菜是某个厨师做的第k道菜，则他的等待时间就是这个厨师制作前k道菜的时间之和。最小化总等待时间。

设同学总数为p, $p \leq 800, n \leq 40, m \leq 100, t_{i,j} \leq 10^3$.

总等待时间比较棘手，可以这样来看：某个厨师做第最后一道菜的时间是$t_{i,j}$，倒数第二道菜的时间为$2t_{i,j}$，倒数第三道为$3t_{i,j}$...这样我们就是最小化做菜的总时间。

NOI2012 美食节

总等待时间比较棘手，可以这样来看：某个厨师做第最后一道菜的时间是$t_{i,j}$，倒数第二道菜的时间为$2t_{i,j}$，倒数第三道为$3t_{i,j}$...这样我们就是最小化做菜的总时间。

对于每个厨师建p个点，第i个点代表这个厨师的倒数第i道菜，并由源点向它们分别连上限为1的边；对于每种菜品建一个点，向汇点连p_i的边。然后第i个厨师的第k个点向第j种菜品连流量为1，费用为$k \times t_{i,j}$的边。答案是满流的最小费用。

NOI2012 美食节

总等待时间比较棘手，可以这样来看：某个厨师做第最后一道菜的时间是$t_{i,j}$，倒数第二道菜的时间为$2t_{i,j}$，倒数第三道为$3t_{i,j}$...这样我们就是最小化做菜的总时间。

对于每个厨师建p个点，第i个点代表这个厨师的倒数第i道菜，并由源点向它们分别连上限为1的边；对于每种菜品建一个点，向汇点连p_i的边。然后第i个厨师的第k个点向第j种菜品连流量为1，费用为$k \times t_{i,j}$的边。答案是满流的最小费用。

但这样复杂度难以接受，这里有另一个技巧：一开始我们只对每个厨师建一个点，增广一次后，会有某个厨师的第一个点被占用，我们再给这个厨师开第二个点。这样点数就大大减少了，复杂度大概是$O(p^2mk)$的。

Thanks