

图论

张尊喆 Hs-black

割点/割边

无向图中讨论

删掉这个点/边，图变得不连通。

tarjan 求割点

随便找个根节点，我们跑 dfs，并且给树打上时间戳 dfn。

对于根节点，我们判断其是否是割点，只需判断是否有大于 1 个子树。

考虑非根节点 x ：

考虑非树边长什么样子：一定是从祖先到子孙！

那么假设 x 不是割点，那么子树内的每个点都能到 x 的上面部分。

设 $low[x]$ 表示 x 子树内，通过非树边走到的最浅是哪里？

如果存在 x 的儿子 y 满足 $low[y] \geq dfn[x]$ ，那么 x 就是割点。

Code

```
int dfn[N], low[N], cut[N], tmp, num, cnt, rt;
void dfs(int x, int fa) {
    dfn[x] = low[x] = ++num;
    for (int i = h[x]; i; i = ne[i]) {
        int y = to[i]; if (y == fa) continue;
        if (!dfn[y]) {
            dfs(y, i ^ 1), low[x] = min(low[x], low[y]);
            if (x != rt && low[y] >= dfn[x]) cut[x] = 1;
            if (x == rt) tmp++;
        }
        else low[x] = min(low[x], dfn[y]);
    }
}
```

点双缩点

```
stk[++st] = u;
for (int i = point[u]; i != -1; i = edge[i].nxt){
    int v = edge[i].v;
    if (!dfn[v]){
        Tarjan(v);
        low[u] = min(low[u], low[v]);
        // 因为v在u的子树内, 所以low[v]可以用于更新low[u]
        if (low[v] >= dfn[u]){ // 子树中没有可以连到父亲上面的边
            cut[u] = 1;
            dccn++;
            dcc[dccn].clear();
            dcc[dccn].push_back(u); // u是割点, 可能包含在多个点双中, 不能弹出
            while (1){
                int w = stk[st];
                dcc[dccn].push_back(w);
                st--;
                if (w == v){ // 做到子树全部弹出为止, 不然v的兄弟也会被弹出
                    break;
                }
            }
        }
    }
}
```

原文链接: <https://blog.csdn.net/xyyxyyx/article/details/83344224>

tarjan 求割边

```
void tarjan(int x, int fa) {
    low[x] = dfn[x] = ++num;
    vis[x] = 1, st[++top] = x;
    int cnt = 0;
    for (int i = h[x]; i; i = ne[i]) {
        int y = to[i];
        if (y == fa && (++cnt) < 2) continue;
        if (!dfn[y])
            tarjan(y, x), low[x] = min(low[x], low[y]);
        else low[x] = min(low[x], dfn[y]);
    }
    if (dfn[x] == low[x] && ++sc)
        while (z = st[top--]) {
            col[z] = sc, vis[z] = 0;
            if (z == x) break;
        }
}
```

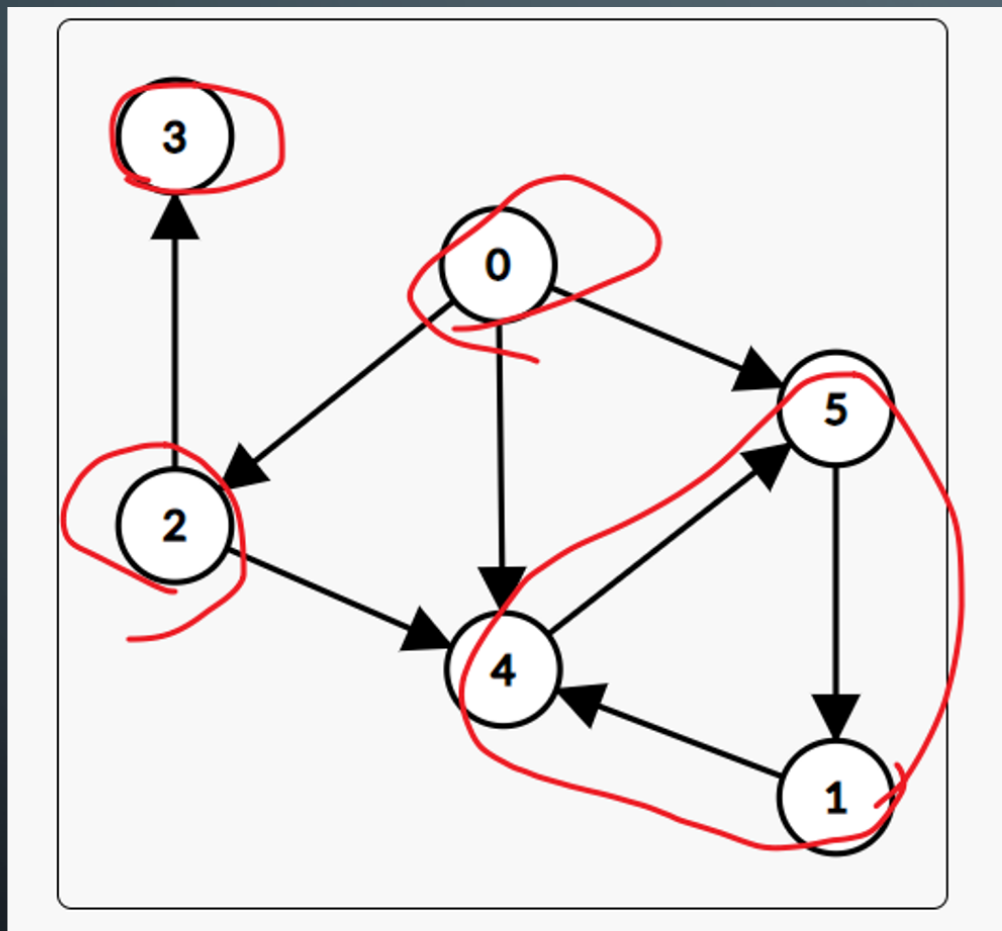
强连通分量

给定有向图，查询有多少对点 u, v 能互相到达。

概念

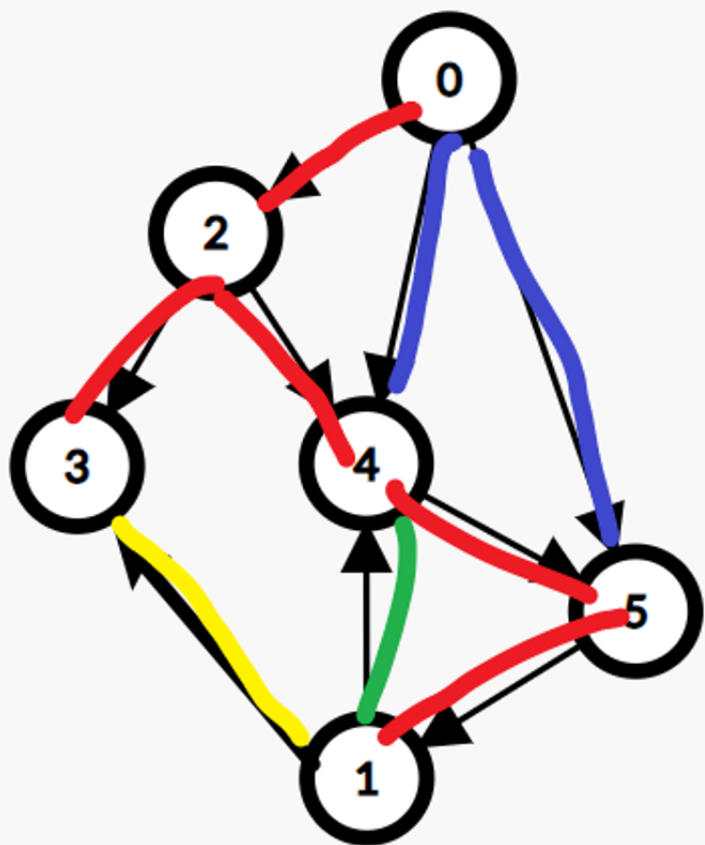
- 在有向图中, 如果两个点 u, v 满足同时存在从 u 到 v 和从 v 到 u 的路径, 则称两个点强连通
- 如果有向图任意两个点强连通, 则称为强连通图. 有向图的极大强连通子图称为强连通分量
- 注意到强连通关系是传递的, 所以有向图可以划分为若干不交的强连通分量

例子

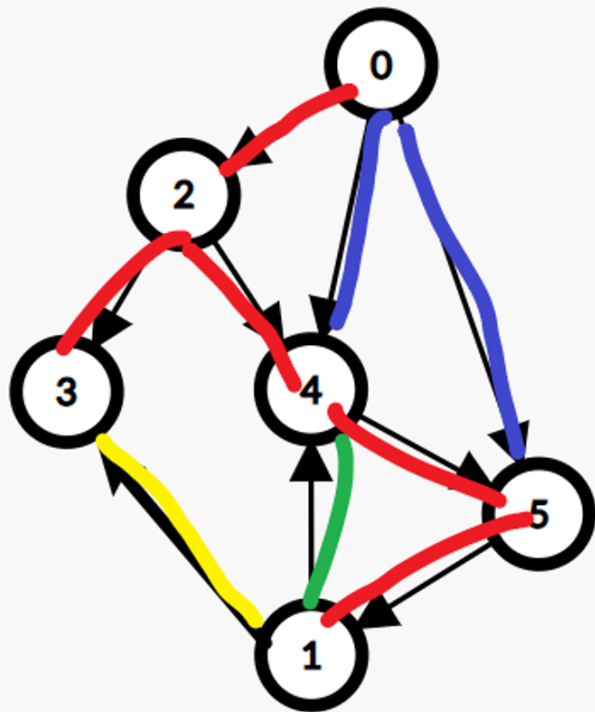


- 如图, 0, 2, 3 各自为一个强连通分量, 1, 3, 4 构成一共强连通分量.

有向图的 DFS 树



- 红色: 树边
- 蓝色: 非树边(完全没用)
- 绿色: 返祖边(有用)
- 黄色: 横叉边(部分有用)



- 蓝色:没用, 比如(0, 4), 代表能从 0 走到 4, 但显然可以走树边到 4, 所以去掉也没事
- 绿色:有用, 比如(1, 4), 代表着 4 到 1 的点构成了一个环, 能互相到达
- 黄色:部分有用, 图中的 (1, 3) 无用, 若添加一条 3 到 2 的边, 那么 (1, 3) 就有用了

- Dfs 到 x 时, 我们用栈维护 x 的每个祖先目前的强连通分量.
- 遍历所有 x 的边 (x, y)
 - 如果 y 未遍历过, Dfs(y)
 - 如果 y 遍历过, 如果 y 在某个祖先的强连通分量中, 则形成了环, 将若干连通分量合并. 否则不做更新
- 合并操作可以通过记录 `low` 数组实现

```
int sta[N], vis[N], num;
void tarjan(int x) {
    low[x] = dfn[x] = ++num;
    sta[++top] = x, vis[x] = 1;
    for (int i = h[x]; i; i = ne[i]) {
        int y = to[i];
        if (!dfn[y]) {
            tarjan(y);
            low[x] = min(low[x], low[y]);
        }
        else if (vis[y]) low[x] = min(low[x], low[y]);
    }
    if (dfn[x] == low[x]) {
        int y;
        while (y = sta[top--]) {
            sd[y] = x;
            vis[y] = 0;
            if (x == y) break;
            w[x] += w[y];
        }
    }
}
```

P3469 [POI2008] BLO-Blockade

B 城有 n 个城镇， m 条双向道路。

每条道路连结两个不同的城镇，没有重复的道路，所有城镇连通。

把城镇看作节点，把道路看作边，容易发现，整个城市构成了一个无向图。

请你对于每个节点 i 求出，把与节点 i 关联的所有边去掉以后（不去掉节点 i 本身），无向图有多少个有序点 (x, y) ，满足 x 和 y 不连通。

$n \leq 100000$, $m \leq 500000$

Solution

魔改一下 tarjan 求割点的过程。

$$\begin{aligned} & sum - \sum_{low[y] \geq dfn[x]} si[y] \\ ans[x] &= (n - sum - 1) \times (sum + 1) + \sum_{low[y] \geq dfn[x]} si[y](n - si[y]) \end{aligned}$$

P1407 [国家集训队]稳定婚姻

我们已知 n 对夫妻的婚姻状况，称第 i 对夫妻的男方为 B_i ，女方为 G_i 。若某男 B_i 与某女 G_j 曾经交往过（无论是大学，高中，亦或是幼儿园阶段， $i \leq j$ ），则当某方与其配偶（即 B_i 与 G_i 或 B_j 与 G_j ）感情出现问题时，他们有私奔的可能性。不妨设 B_i 和其配偶 G_i 感情不和，于是 B_i 和 G_j 旧情复燃，进而 B_j 因被戴绿帽而感到不爽，联系上了他的初恋情人 G_k 一串串的离婚事件像多米诺骨牌一般接踵而至。若在 B_i 和 G_i 离婚的前提下，这 $2n$ 个人最终依然能够结合成 n 对情侣，那么我们称婚姻 i 为不安全的，否则婚姻 i 就是安全的。

给定所需信息，你的任务是判断每对婚姻是否安全。

$1 \leq n \leq 4000$, $0 \leq m \leq 20000$ 。

Solution

婚姻让女方向男方连边，旧情让男方向女方连边。不安全的婚姻说明出现了环，也就是在一个连通分量内。

所以我们跑 tarjan，一对夫妻在强连通分量里面说明不安全。

UOJ67. 新年的毒瘤

辞旧迎新之际，喜羊羊正在打理羊村的绿化带，然后他发现了一棵长着毒瘤的树。

这个长着毒瘤的树可以用 n 个结点 m 条无向边的无向图表示。这个图中有一些结点被称作是毒瘤结点，即删掉这个结点和与之相邻的边之后，这个图会变为一棵树。树也即无简单环的无向连通图。

现在给你这个无向图，喜羊羊请你帮他求出所有毒瘤结点。

$$1 \leq n, m \leq 10^5$$

Solution

只需满足两个条件

- 删掉的点不能是割点
- 删掉这个点以后满足点数等于边数 + 1

P4652 [CEOI2017] One-Way Streets

给定一张 n 个点 m 条边的无向图，现在想要把这张图定向。

有 p 个限制条件，每个条件形如 (x_i, y_i) ，表示在新的有向图当中， x_i 要能够沿着一些边走到 y_i 。

现在请你求出，每条边的方向是否能够唯一确定。同时请给出这些能够唯一确定的边的方向。

数据保证有解。

对于所有测试点，有 $1 \leq n, m, p \leq 100\,000$; $1 \leq a_i, b_i, x_i, y_i \leq n$ 。

Solution

因为保证有解。我们考虑原图中边双里面的边无法确定方向，所以考虑缩边双。

缩完以后，现在相当于有一棵树的原问题。

可以树上差分，比如从 x 到 y ，让 $d[x]++$, $d[y]--$ ，这样边权为正代表向上，边权为负代表向下。

CF402E Strictly Positive Matrix

给出一个矩阵 A , 问是否存在一个正整数 k 使得 A^k 的所有元素都是正数。

$$2 \leq n \leq 2000, 0 \leq a_{i,j} \leq 50, \sum_{i=1}^n a_{i,i} > 0$$

Solution

矩阵乘法转化为计算路径的条数。

即询问是否存在 k 使得所有 i 到 j 都有长度为 k 的路径。

那么需要所有人都在一个强连通分量里面，注意存在某个点有自环，所以可以凑出长度为 k 的路径。

P3225 [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。

请写一个程序，用来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。

$$n \leq 500, m \leq 1000$$

Solution

将点双缩点变成一棵树，那么叶子节点一定要建，非叶子节点可以不用建。

单独讨论整个图是一个点双的情况。

二分图

P1155 [NOIP2008 提高组] 双栈排序

给你一个长为 n 的序列 p , 问是否能够通过对于两个栈进行 push, pop(print) 操作使得最后输出序列单调递增 (即为 $1 \cdots n$), 如果无解输出 0。

每个操作有个优先级, $\text{push}(1) > \text{pop}(1) > \text{push}(2) > \text{pop}(2)$, 输出优先级最大的一组解。

$$n \leq 1000$$

Solution

当只有一个栈的时候，怎样是合法的？

如果存在 $i < j < k$, $a_k < p_i < p_j$ 不合法。

对于两个栈的情况，我们找到所有不合法的 i, j ，这两个人不能放在一起，想到二分图染色。

找到字典序最小的合法解。

二分图最大匹配

一定存在增广路，扩展所有增广路即可。

二分图最小点覆盖

左侧未匹配点开始 dfs，答案为左侧未 dfs 到的点 + 右侧 dfs 到的点。

二分图最大独立集

最小点覆盖的补集

P2055 [ZJOI2009] 假期的宿舍

学校放假了.....有些同学回家了，而有些同学则有以前的好朋友来探访，那么住宿就是一个问题。

比如 A 和 B 都是学校的学生，A 要回家，而 C 来看 B，C 与 A 不认识。我们假设每个人只能睡和自己直接认识的人的床。那么一个解决方案就是 B 睡 A 的床而 C 睡 B 的床。而实际情况可能非常复杂，有的人可能认识好多在校学生，在校学生之间也不一定都互相认识。

我们已知一共有 n 个人，并且知道其中每个人是不是本校学生，也知道每个本校学生是否回家。问是否存在一个方案使得所有不回家的本校学生和来看他们的其他人都有地方住。

对于 100% 的数据满足 $1 \leq n \leq 50, 1 \leq T \leq 20$ 。

P4304 [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。

对于 100% 的数据，保证 $N \leq 200$ 。

Solution

对棋盘黑白染色，有攻击关系建边。那么就是二分图最大独立集 = n - 最大匹配。

P1129 [ZJOI2007] 矩阵游戏

小 Q 是一个非常聪明的孩子，除了国际象棋，他还很喜欢玩一个电脑益智游戏——矩阵游戏。矩阵游戏在一个 $n \times n$ 黑白方阵进行（如同国际象棋一般，只是颜色是随意的）。每次可以对该矩阵进行两种操作：

- 行交换操作：选择矩阵的任意两行，交换这两行（即交换对应格子的颜色）。
- 列交换操作：选择矩阵的任意两列，交换这两列（即交换对应格子的颜色）。

游戏的目标，即通过若干次操作，使得方阵的主对角线(左上角到右下角的连线)上的格子均为黑色。

对于某些关卡，小 Q 百思不得其解，以致他开始怀疑这些关卡是不是根本就是无解的！于是小 Q 决定写一个程序来判断这些关卡是否有解。

- 对于 100% 的数据，保证 $1 \leq n \leq 200$, $1 \leq T \leq 20$ 。

Solution

行和列都建点跑二分图匹配。

P1263 [CEOI2002] Royal guards

从前有一个王国，这个王国的城堡是 m 行 n 列的一个矩形，被分为 $m \times n$ 个方格。一些方格是墙，而另一些是空地。这个王国的国王在城堡里设了一些陷阱，每个陷阱占据一块空地。

一天，国王决定在城堡里布置守卫，他希望安排尽量多的守卫。

守卫们都是经过严格训练的，所以一旦他们发现同行或同列中有人的话，他们立即向那人射击。因此，国王希望能够合理地布置守卫，使他们互相之间不能看见，这样他们就不可能互相射击了。守卫们只能被布置在空地上，不能被布置在陷阱或墙上，且一块空地只能布置一个守卫。如果两个守卫在同一行或同一列，并且他们之间没有墙的话，他们就能互相看见。(守卫就像象棋里的车一样)

你的任务是写一个程序，根据给定的城堡，计算最多可布置多少个守卫，并设计出布置的方案。

对于全部的测试点，保证 $1 \leq m, n \leq 200$, $0 \leq a_{i,j} \leq 2$ 。

Solution

将连续的一段空地而不是一整行建成一个点，类似矩阵游戏的连边。

基环树

P1453 城市环路

整个城市可以看做一个 n 个点, n 条边的单圈图(保证图连通), 唯一的环便是绕城的环路。保证环上任意两点有且只有 2 条简单路径互通。图中的其它部分皆隶属城市郊区。

现在, 有一位名叫 Jim 的同学想在 B 市开店, 但是任意一条边的 2 个点不能同时开店, 每个点都有一定的人流量, 第 i 个点的人流量是 p_i , 在该点开店的利润就等于 $p_i \times k$, 其中 k 是一个常数。

Jim 想尽量多的赚取利润, 请问他应该在哪些地方开店?

- 对于 100% 的数据, 保证 $1 \leq n \leq 10^5$, $1 \leq p_i \leq 10^4$, $0 \leq u, v < n$, $0 \leq k \leq 10^4$, k 的小数点后最多有 6 位数字。

Solution

对于树来说，发现就是求最大独立集，设 $f[x][0/1]$ 表示 x 点选或不选的最大值。

对于环来说，考虑断环为链，相邻两个点一定有一个是不选，分别枚举强制某个点不选做一次链上的 DP 即可。

P4381 [IOI2008] Island

给出一个基环树森林（每一个点有一条无向边），有正边权，让你求出所有基环树的直径（即一条不经过重复点的基环树上的最长路径）之和。

对于 100% 的数据， $2 \leq N \leq 10^6$ 。

Solution

考虑求一个基环树的直径，先求出树的直径。

对于环上的每个点，我们考虑记录 $f[x]$ 表示从 x 开始到其子树内最长路径是多少。

断环为链，记录链上的路径前缀和，那么答案就是 $\max(f_i + f_j + s_j - s_i)$,
 $\max(f_i + f_j + len - (pre_j - pre_i))$, 代表走环的两部分。

直接扫一遍即可。

P5049 [NOIP2018 提高组] 旅行 加强版

城市构成一棵树或者一棵基环树。

小 Y 的旅行方案是这样的：任意选定一个城市作为起点，然后从起点开始，每次可以选择一条与当前城市相连的道路，走向一个没有去过的城市，或者沿着第一次访问该城市时经过的道路后退到上一个城市。当小 Y 回到起点时，她可以选择结束这次旅行或继续旅行。需要注意的是，小 Y 要求在旅行方案中，每个城市都被访问到。

为了让自己的旅行更有意义，小 Y 决定在每到达一个新的城市（包括起点）时，将它的编号记录下来。她知道这样会形成一个长度为 n 的序列。她希望这个序列的字典序最小，你能帮帮她吗？

对于 100% 的数据和所有样例， $1 \leq n \leq 500000$ 且 $m = n - 1$ 或 $m = n$ 。