动态规划

清华大学 张华清

引入

- 给定一张图，从u到v有mp[u][v]条有向边。
- 求走恰好k条边，从u走到v的方案数？

引入

- 考虑DP:ans[t][x][y]为走了t条边，从x走到y的方案数。
- $ans[t][x][y]=\sum_{w=1}^{n} ans[t-1][x][w]*mp[w][y]$
- 还是枚举中间点嘛。
- 发现可以写成矩阵乘法的形式：$Ans_t=Ans_{t-1}*Mp$
- 那写下去？
- $Ans_t=Ans_0*Mp^t$
- Ans0是单位矩阵。所以$Ans_t=Mp^t$

引入

- 类似整数的快速幂，可以进行矩阵快速幂。
- 所以原问题从tn^3优化到了$n^3\log t$。

6569 [NOI Online #3 提高组] 魔法值

题目描述

复制Markdown 展开

H 国的交通由 n 座城市与 m 条道路构成，城市与道路都从 1 开始编号，其中 1 号城市是 H 国的首都。H 国中一条道路将把两个不同城市直接相连，且任意两个城市间至多有一条道路。

H 国是一个信奉魔法的国家，在第 j 天，i 号城市的魔法值为 $f_{i,j}$。H 国的魔法师已观测到第 0 天时所有城市的魔法值 $f_{i,0}$，且他们还发现，之后的每一天每个城市的魔法值，都将会变为所有与该城市直接相连的城市的前一天魔法值的异或值，即

$$f_{x,j} = f_{v_1,j-1} \oplus f_{v_2,j-1} \oplus \cdots \oplus f_{v_k,j-1}$$

其中 $j \geq 1$，$v_1, v_2, \cdots, v_k$ 是所有与 x 号城市直接相连的城市，$\oplus$ 为异或运算。

现在 H 国的国王问了你 q 个问题，对于第 i（$1 \leq i \leq q$）个问题你需要回答：第 a_i 天时首都的魔法值是多少。

说明/提示

数据规模与约定

- 对于 20% 的数据，满足 $a_i \leq 100$。
- 对于 40% 的数据，满足 $n \leq 20$。
- 另有 30% 的数据，满足 $m = \frac{n(n-1)}{2}$。
- 对于 100% 的数据，满足 $1 \leq n, q \leq 100$，$1 \leq m \leq \frac{n(n-1)}{2}$，$1 \leq a_i < 2^{32}$，$0 \leq f_i < 2^{32}$。

6569 [NOI Online #3 提高组] 魔法值

- 每个1是通过一条路径传给首都的。
- 也就是说，x这个点到首都有几条长恰好为ai的路径，首都就会被异或几次f_i。
- 于是我们只需要关心所有点到首都长为a_i路径条数的奇偶性。如果为奇数则答案异或上f_i。
- 怎么求每个点到首都长为a_i的路径条数？这不是我们讲过的问题嘛！
- 但是有多组询问欸，直接矩阵快速幂复杂度q*n^3loga，好像过不去

6569 [NOI Online #3 提高组] 魔法值

- 法一：求方案数的奇偶性，也即对2取模。
- %2意义下，乘法相当于&，加法相当于^。
- 可以使用bitset优化。
- 即A矩阵的第x行和B矩阵的第y列&起来，再看看几个1。奇数个ans[x][y]就为1，否则为0。

6569 [NOI Online #3 提高组] 魔法值

- 法二：多组询问的经典套路。
- 我们要求一个Ans_k=Ans_0*Mp^k，其中Ans_i是一个长度为n的行向量，表示从1出发，走恰好i步走到x这个点的方案数%2.
- 就我们是在求一个行向量和log k个n*n的矩阵的成绩。
- 那么，我们先预处理Mp^k。
- 然后，算答案的时候，我们别先算这log k个方阵的乘积啊！
- 我们就用Ans_0向量从左乘到右！因为向量乘矩阵复杂度是O(n^2)的！
- 这样复杂度就从q*n^3logt，变成了n^3logt+q*n^2logt

6569 [NOI Online #3 提高组] 魔法值

- 两个优化方法使用其一即可过。
- 两个都用巨大快。

NOI2020 美食家

题目描述

复制Markdown 展开

坐落在 Bzeroth 大陆上的精灵王国击退地灾军团的入侵后，经过十余年的休养生息，重新成为了一片欣欣向荣的乐土，吸引着八方游客。小 W 是一位游历过世界各地的著名美食家，现在也慕名来到了精灵王国。

精灵王国共有 n 座城市，城市从 1 到 n 编号，其中城市 i 的美食能为小 W 提供 c_i 的愉悦值。精灵王国的城市通过 m 条**单向道路**连接，道路从 1 到 m 编号，其中道路 i 的起点为城市 u_i ，终点为城市 v_i，沿它通行需要花费 w_i 天。也就是说，若小 W 在第 d 天从城市 u_i 沿道路 i 通行，那么他会在第 $d+w_i$ 天到达城市 v_i。

小 W 计划在精灵王国进行一场为期 T 天的旅行，更具体地：他会在第 0 天从城市 1 出发，经过 T 天的旅行，最终在**恰好第 T 天**回到城市 1 结束旅行。由于小 W 是一位美食家，每当他到达一座城市时（包括第 0 天和第 T 天的城市 1），他都会品尝该城市的美食并获得其所提供的愉悦值，若小 W 多次到达同一座城市，他将**获得多次愉悦值**。注意旅行途中小 W **不能在任何城市停留**，即当他到达一座城市且还未结束旅行时，他当天必须立即从该城市出发前往其他城市。

对于所有测试点：

$1 \le n \le 50$，$n \le m \le 501$，$0 \le k \le 200$，$1 \le t_i \le T \le 10^9$。

$1 \le w_i \le 5$，$1 \le c_i \le 52501$，$1 \le u_i, v_i, x_i \le n$，$1 \le y_i \le 10^9$。

每个测试点的具体限制见下表：

测试点编号	n	m	T	特殊限制
$1 \sim 4$	≤ 5	≤ 50	≤ 5	无
$5 \sim 8$	≤ 50	≤ 50	≤ 52501	无
$9 \sim 10$	≤ 50	≤ 50	$\le 10^9$	A
$11 \sim 13$	≤ 50	≤ 50	$\le 10^9$	$k = 0$
$14 \sim 15$	≤ 50	≤ 50	$\le 10^9$	$k \le 10$
$16 \sim 17$	≤ 50	≤ 50	$\le 10^9$	无
$18 \sim 20$	≤ 50	≤ 501	$\le 10^9$	无

特殊限制 A：$n = m$ 且 $u_i = i, v_i = (i \bmod n) + 1$。

NOI2020 美食家

- 如果所有边长都是1，还没有美食节，那做邻接矩阵A的k次幂（此处的矩阵乘法为+,max矩阵乘法）不就是最大收益了嘛
- 但是有问题:
- 有边长怎么办?
  - 有特殊性质：边长很短
  - 法一：每条边拆点。无法通过所有的数据
  - 法二：想想怎样用矩阵乘法求斐波那契数列，我们开了一维表示f[i-1]。这里我们也对每个点另开4个分身记录上一个，2个，3个，4个时刻的状态。即f[t][x],f[t-1][x],f[t-2][x],f[t-3][x],f[t-4][x]。可以通过本题数据

NOI2020 美食家

- 有美食节怎么办？拆成若干段分别求。
- 每次相当于一个新的询问。相当于询问k+1次
- 和上题一样的经典操作！
- 我们其实不想知道x到y的答案
- 只想知道从1出发的答案
- 所以只需要倍增预处理后用一个1*n的向量乘过去。
- 复杂度n^3logT+kn^2logT

SP1716 GSS3 - Can you answer these queries III

n 个数，q 次操作

操作 `0 x y` 把A_x 修改为y

操作 `1 l r` 询问区间$[l,r]$ 的最大子段和

SP1716 GSS3 - Can you answer these queries III

- 先考虑怎么用DP写：
- f[i]为以i结尾的最大子段和是多少。
- 则f[i]=max(f[i-1]+v[i],v[i])
- 设g[i]为1...i中，最大子段和为多少。则g[i]是f[i]的前缀最大值，即：
- g[i]=max(g[i-1],f[i])

SP1716 GSS3 - Can you answer these queries III

- 怎么写成广义矩阵乘法的形式?
- v[i]放在转移矩阵里。分别考虑f[i-1],g[i-1]对f[i],g[i]的贡献就行啦!
- 有点困难？困难在单独的v[i]咋整?
- 单独在行向量里加一个元素——0
- 即考虑一个行向量Ai=[f[i],g[i],0]，考虑A[i-1]怎么转移到A[i]即可

SP1716 GSS3 - Can you answer these queries III

- 实现上：线段树上每个叶子节点就存一个转移矩阵。
- 其他节点，存它这个区间的矩阵从左到右乘起来是啥。
- 然后询问时，查询线段树上[l,r]的转移矩阵的成绩。就像回答区间和那样就可以啦！
- 然后用初始向量[0,-INF,0]去乘一下，发现答案就是max(A[1][2],A[3][2])（下标从1计数。

Luogu 1655 小朋友的球

题目描述

复制Markdown 展开

@发源于 小朋友最近特别喜欢球。有一天他脑子抽了，从口袋里拿出了 N 个不同的球，想把它们放到 M 个相同的盒子里，并且要求每个盒子中至少要有一个球，他好奇有几种放法，于是尝试编程实现，但由于他天天不好好学习，只会上 B 站看游泳教练，于是他向你求助。

$n \le 100$

$n, m \le 1000$

m

Luogu 1655 小朋友的球

- 有“球”和“盒子”两个元素，所以我们会想到设两维状态
- 设f[i][j]为i个球，放到j个盒子里的方案数。
- 讨论最后一个球，它是自己一个盒子，还是和其他的球一个盒子。
- 如果是它自己一个盒子的话，那好说，前i-1个球就只能放在j-1个盒子里，方案数就是f[i-1][j-1]
- 如果它和某些个球放在一个盒子里，方案数好像并不是f[i-1][j]诶！因为咱们还得考虑，它到底是放在哪一个盒子里？所以好像还应该乘个j诶！

Luogu 1655 小朋友的球

- 综上f[i][j]=f[i-1][j-1]+j*f[i-1][j]
- ——第二类斯特林数

$\cancel{f[i-1][j]}$

$$f[n][m] = \sum_{i=1}^{n} \binom{n-1}{i-1} f[n-i][m-1]$$

Luogu 1655 小朋友的球

- 综上f[i][j]=f[i-1][j-1]+j*f[i-1][j]
- ——第二类斯特林数

6189 [NOI Online #1 入门组] 跑步

n≤1e5.?

n≤1e3?

题目描述

复制Markdown 展开

小 H 是一个热爱运动的孩子，某天他想给自己制定一个跑步计划。小 H 计划跑 n 米，其中第 $i(i \geq 1)$ 分钟要跑 x_i 米（x_i 是正整数），但没有确定好总时长。

由于随着跑步时间增加，小 H 会越来越累，所以小 H 的计划必须满足对于任意 $i(i > 1)$ 都满足 $x_i \leq x_{i-1}$。

?

现在小 H 想知道一共有多少个不同的满足条件的计划，请你帮助他。两个计划不同当且仅当跑步的总时长不同，或者存在一个 i，使得两个计划中 x_i 不相同。

由于最后的答案可能很大，你只需要求出答案对 p 取模的结果。

6189 [NOI Online #1 入门组] 跑步

- 整数拆分问题：给定n，求出把n分解成若干无序整数的和的方案数。
- 注意：题意中$x_i \le x_{i+1}$就等价于拆分出的整数是无序的。
- 这也相当于球本质相同，盒子本质相同，盒子数量任意的球盒问题。

3 = 1+1+1

= 1+2

= 2+1 → 归 → 一种方案

6189 [NOI Online #1 入门组] 跑步

$0 \le x_i \le K$

- 问题相当于完全背包，物品体积分别为1,2,3...n，可以用无限个，需要填满大小为n的背包。
- 因而有DP：设f[i][j]为用<=i的物品，拼出j的方案数。那么f[i][j]=f[i-1][j]+f[i][j-i]。
- *这个Dp方式可以求，x_i有上界的方案数。*

$K=1+1+1+1$

$3=1+2$

6189 [NOI Online #1 入门组] 跑步

$O(n^2)$

- 但毕竟这和完全背包还是有不一样的地方。
- 我们还可以这么设状态：g[i][j]为用了i个物品，凑出j的方案数。
- 转移也比较神奇，讨论是否有1。
- 如果没有1，从g[i][j-i]转移。
- 如果有1，从g[i-1][j-1]转移。
- 即g[i][j]=g[i][j-i]+g[i-1][j-1]。
- *这个Dp方式可以求，最多（恰好）分成几份的方案数。*

6189 [NOI Online #1 入门组] 跑步

$\le x_1 \le x_2 \le x_3 \cdots$

- 挖掘一下，这个DP还可以**钦定最小元素!**
- 比如你钦定最小元素是m，那你就讨论整个方案中有没有m。
- 如果没有，全员-1，所有数还是>=m，从g[i][j–i]转移。
- 如果有，直接刨掉，从g[i-1][j-m]转移。
- 即g[i][j]=g[i–1][j–m]+g[i][j–i]。

6189 [NOI Online #1 入门组] 跑步

$O(n\sqrt{n})$

- 现在我们有两种方法了但似乎都不能通过本题….
- 根号分治！
- 对于小于等于$B=\sqrt{n}$的物品，用第一种背包的求法。
- 对于大于B的物品，用第二种求法（因为可以钦定最小物品）。这样用的物品数不超过$\sqrt{n}$。
- 这样两部分DP都是$O(n\sqrt{n})$的复杂度，可以接受。
- 最后卷积求出总方案数。（枚举小于等于B的物品拼出了i，那么大于B的物品拼出了n-i，乘起来再对i求和。）

$x_i \le B$

$x_i \ge B+1$

$B\cdot n + \frac{n^2}{B}$

$B=\sqrt{n}$

$n\sqrt{n}$

答案 $\sum_{i=0}^{n} f_i \times g_{n-i}$

Luogu 1450 [HAOI2008] 硬币购物

题目描述

复制Markdown 展开

共有 4 种硬币。面值分别为 c_1, c_2, c_3, c_4。

某人去商店买东西，去了 n 次，对于每次购买，他带了 d_i 枚 i 种硬币，想购买 s 的价值的东西。请问每次有多少种付款方法。

说明/提示

数据规模与约定

- 对于 100% 的数据，保证 $1 \le c_i, d_i, s \le 10^5$，$1 \le n \le 1000$。

Luogu 1450 [HAOI2008] 硬币购物

$S_i = \sum_{j=0}^{n} \binom{n}{j} a_j$

$a_i = \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} S_j$

$S_i = \sum a_i$

- 若跑n遍多重背包，复杂度无法接受。
- 使用 没有任何限制的总方案数 – 有某一个限制一定被打破，其他限制无所谓+ 有某两个限制一定被打破，其他无所谓 + 有某三个限制一定被打破，其他无所谓 – 所有限制都被打破。
- 没有任何限制的总方案数？完全背包预处理，设f[x]为用任意数量的四种钱币购买x价值的方案数。即为f[s]
- 有一个限制被打破？ f[s-c[1]*(d[1]+1)], f[s-c[2]*(d[2]+1)]…
- 有两个限制被打破？ f[s-c[1]*(d[1]+1)-c[2]*(d[2]+1)]…

$S_i - S_{i-1}$

$S_i = \sum_{j=1}^{n} a_j$

反演

$a_i = S_i - S_{i-1}$

Luogu 1450 [HAOI2008] 硬币购物

```
while(tot--){
    for(j=0;j<4;++j)scanf("%d",&d[j]);
    scanf("%d",&s);
    ans=f[s];
    for(ss=1;ss<=15;++ss){//二进制数枚举集合，容斥
        now=s;
        for(tmp=ss,j=k=0;tmp;tmp>>=1,++j)
            if(tmp&1)k^=1,now-=(d[j]+1)*c[j];
            //注意k的作用，判断奇偶
        if(now>=0)k?ans-=f[now]:ans+=f[now];
    }
    printf("%lld\n",ans);
}
```

4099 [HEOI2013] SAO

- 题意：求树形有向图的拓扑序个数。n<=1000

4099 [HEOI2013] SAO

- 先考虑外向树。
- 拓扑序的要求：每个点x都是其子树中第一个出现的。
- 各个子树独立。则合法拓扑序个数为$\frac{n!}{\prod siz[x]}$，其中siz[x]为x子树大小。

4099 [HEOI2013] SAO

- 再考虑有反向边的情形。
- 有正有反不好做，考虑容斥。
- 把每一条反向边拆成：（没有任何限制-有正向边）。
- 则总方案数为：所有反向边都不存在的方案数-某一条反向边改为正向边，其他反向边不存在+某两条反向边改为正向边，其他反向边不存在+…
- 我们当然可以2^n枚举每条反向边是否存在，得到的结果即外向树森林，答案还是$\frac{n!}{\prod siz[x]}$。

4099 [HEOI2013] SAO

- 容斥DP！——把系数放到DP过程中。
- 只需要记录所需的额外信息，本题中即为x所在子树大小（即把所有“当作不存在”的反向边删去后的子树大小）。
- 设f[x][i]为x子树内，x所在子树大小为i的值。
- 对于一条正向边，像树形背包那样正常合并。
- 对于一条反向边，分1*没有任何限制和-1*变成正向边转移。
- 复杂度同树形背包，为$O(n^2)$。

2761 软件补丁问题

dp[i][S]:

前i个补丁

= min(dp[i-1][S']

题目描述

复制Markdown 展开

T 公司发现其研制的一个软件中有 n 个错误，随即为该软件发放了 m 个补丁程序。

每一个补丁程序都有其特定的适用环境，某个补丁只有在软件中包含某些错误而同时又不包含另一些错误时才可以使用。一个补丁在排除某些错误的同时，往往会加入另一些错误。

换句话说，对于任意一个补丁 i，都有四个与之相应的集合 $B1_i, B2_i, F1_i$ 和 $F2_i$。仅当软件包含 $B1_i$ 中的所有错误，而不包含 $B2_i$ 中的任何错误时，才可以使用补丁 i。补丁 i 将修复软件中的某些错误集合 $F1_i$，而同时加入另一些错误 $F2_i$。另外，运行每个补丁都耗费一定的时间。

试设计一个算法，利用 T 公司提供的 m 个补丁程序将原软件修复成一个没有错误的软件，并使修复后的软件耗时最少。对于给定的 n 个错误和 m 个补丁程序，找到总耗时最少的软件修复方案。

说明/提示

对于 100% 的数据：$1 \le n \le 20$，$1 \le m \le 100$。

2761 软件补丁问题

- 我们建立2^n个点。编号看成一个n位的二进制数。所有1的位置代表“含有第i个bug”，0的位置代表“不含有第i个bug”。
- 如，n=3时，100代表只含有3号bug，011代表含有1号和2号bug。
- 对于每个点S，我们枚举每一个补丁i，如果S可以用i号补丁，即S包含B1[i]中的所有错误，而不包含B2[i]中的任何错误，我们就连边S->T，其中T是S使用了i补丁后的结果（即，把所有F1[i]的bug修好（1->0），把所有F2[i]的bug加上（0->1））。

2761 软件补丁问题

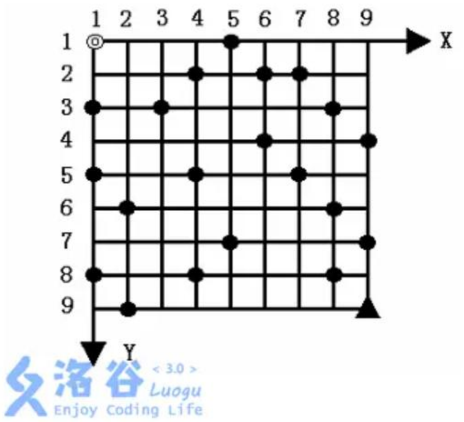
- 注：这可以理解成是一个动态规划问题，但转移成环了。这时我们就需要使用最短路算法，而不是按某个顺序DP。（通常的动态规划问题的转移形成一张DAG，DP的顺序就是其某个拓扑序。）

4009 汽车加油行驶问题

题目描述

复制Markdown 展开

给定一个 $N \times N$ 的方形网格，设其左上角为起点◎，坐标$(1,1)$，X 轴向右为正， Y 轴向下为正，每个方格边长为 1 ，如图所示。



一辆汽车从起点◎出发驶向右下角终点▲，其坐标为 (N,N)。

在若干个网格交叉点处，设置了油库，可供汽车在行驶途中加油。汽车在行驶过程中应遵守如下规则:

1. 汽车只能沿网格边行驶，装满油后能行驶 K 条网格边。出发时汽车已装满油，在起点与终点处不设油库。
2. 汽车经过一条网格边时，若其 X 坐标或 Y 坐标减小，则应付费用 B ，否则免付费用。
3. 汽车在行驶过程中遇油库则应加满油并付加油费用 A。
4. 在需要时可在网格点处增设油库，并付增设油库费用 C(不含加油费用A)。
5. N,K,A,B,C 均为正整数， 且满足约束: $2 \le N \le 100, 2 \le K \le 10$。

设计一个算法，求出汽车从起点出发到达终点所付的最小费用。

说明/提示

$2 \le n \le 100, 2 \le k \le 10$

4009 汽车加油行驶问题

- 显然设f[x][i]表示到x这个点剩余油量为k的最小花费。
- 转移成环，使用最短路算法求解。
- 我们可以新建加油站，但下次经过同一点时，怎么知道建没建过加油站呢？
- ——其实不需要知道！
- 你建一次加油站一定会加满油，那之后就不可能再走回这个点了：如果重复经过一个点，那第二次的油量一定比第一次多（比如绕路加个油啥的），否则绕这一圈没有意义。

Luogu 4547 [THUWC2017] 随机二分图

题目描述

某人在玩一个非常神奇的游戏。这个游戏中有一个左右各 n 个点的二分图，图中的边会按照一定的规律随机出现。

为了描述这些规律，某人将这些边分到若干个组中。每条边或者不属于任何组 （这样的边一定不会出现），或者只属于一个组。

有且仅有以下三类边的分组：

0. 这类组每组只有一条边，该条边恰好有 50% 的概率出现。
1. 这类组每组恰好有两条边，这两条边有 50% 的概率同时出现，有 50% 的概率同时不出现。
2. 这类组每组恰好有两条边，这两条边恰好出现一条，各有 50% 的概率出现。

组和组之间边的出现都是完全独立的。

某人现在知道了边的分组和组的种类，想要知道完美匹配数量的期望是多少。你能帮助她解决这个问题吗?

【数据规模和约定】

对于 5% 的数据 $n \le 5$ 。
对于另 5% 的数据 $n \le 8$ 。
对于另 10% 的数据 $n \le 10$ 。
对于另 15% 的数据，只有$t = 0$ 的情况。
对于另 5% 的数据，只有$t = 0$ 的情况，且$m = n^2$，也就是该图为一个完全图。
对于另 20% 的数据，只有 $t = 0$ 或者 $t = 1$ 的情况。 对于另 20% 的数据，只有 $t = 0$ 或者 $t = 2$ 的情况。 对于 100% 的数据，$n \le 15$。

Luogu 4547 [THUWC2017] 随机二分图

- 暴力1：别枚举每种连边状态，然后算最大匹配数。这样复杂度$O(2^m 2^n)$。或许能拿5pts?
- 暴力2：枚举所有可能的最大匹配（也即枚举全排列），算贡献。即多少种情况中，该最大匹配中所有的边都存在。复杂度$\mathrm{O}(n! \cdot n)$，期望得分10pts。

Luogu 4547 [THUWC2017] 随机二分图

- 正解沿用了暴力2算贡献的想法。
- 我们显然还是希望用状压的方法，把$n!$变为2^n。比如说，我们想记f[S]为左边前|S|个结点，匹配了右边S集合中的点，所得到的概率和。但困难之处在于，之前一条边的选择与否，会影响同组的边是否被选择。

Luogu 4547 [THUWC2017] 随机二分图

- 有了大致思路后，我们先从简单情况想起。
- 只有0类组?
- 我们设dp[S]为，左边前|S|个结点匹配右边的S集合中结点，能够发生的概率和。
  - 更具体地，它表示：对于所有用前|S|个结点匹配S中结点的方案，考虑随机生成的图使得该方案合法（也即该方案中所有的匹配边都存在）的概率，的和。
- 转移是简单的：枚举左边第|S|号结点匹配谁，乘上1/2转移过去即可。
- Q：我们为什么DP概率和而不DP方案数?
- A：概率和只和方案内的边有关，好处理；而方案数和外界有关，不好处理。

Luogu 4547 [THUWC2017] 随机二分图

- 有了1类组和2类组会有什么问题？组内先被DP到的那条边的选择与否，会影响组内第二条边的选择与否——有后效性了。但我们又不能直接把每个组的第一条边是否选择记到DP状态里。
- 同时请注意“在图中”和“是匹配边”是两回事。如第2组边，只是说他们同时在图中，不是说他们同时要成为匹配边。

Luogu 4547 [THUWC2017] 随机二分图

- 先只考虑第0类和第1类边
- 考虑一组匹配方案，它应该贡献1/2的几次方的概率和？该方案内有多少**组**边！
- 如果是如果我们就把第1类组里的两条边当成两个独立的边，会发生什么？
- 如果匹配方案中只选中了其中的一条边，那确实就贡献了1/2，很合理。
- 如果匹配方案中同时选中了其中的两条边，那就贡献了${\frac{1}{2}}^2$的系数，但应该只贡献一个$\frac{1}{2}$。那咋办？容斥掉！额外加入一组转移，即同时选中两条边，其转移系数为$\frac{1}{4}$。那么同时选中两条边的case就会从以上两种方式转移来，其系数的和就是我们想要的$\frac{1}{2}$.

Luogu 4547 [THUWC2017] 随机二分图

- 同理，对于第2类边，如果一个匹配方案选中了两条边中的一条，那系数为1/2，没问题。但如果同时选中两个，贡献系数为$\frac{1}{2^2}$,但本应为0！
- 额外加入一组转移，即同时选中两条边，其转移系数为-$\frac{1}{4}$。那么同时选中两条边的case就会从以上两种方式转移来，其系数的和就是我们想要的$\frac{1}{2}$.

Luogu 4547 [THUWC2017] 随机二分图

- 实现细节：需要同时记录左边哪些点匹配了，右边哪些点匹配了。即f[S][T]。但总状态数不是(2^n)^2，而是$\sum_{i=0}^{n} C(n,i)^2 = C(2n,n)$。1e8左右，用哈希表存状态。
- 转移时为了不因选边顺序不同重复统计，钦定每次从S集合中编号最小的点出发进行转移。

谢谢大家！