缩圈，路径压缩，状态压缩技巧

demerzel

2018 年 8 月 10 日

- 你们给我搞的这个讲课内容啊，exciting!
- 那么今天上午主要讲一些状压题。
- 那么另外两个呢，我可以回答一句“无可奉告”。

- 带花树? 

- 带花树？ 
- 当然不是……

定义如下：

在有向图 G 中，如果两个顶点 v_i，v_j 间（$v_i > v_j$）有一条从 v_i 到 v_j 的有向路径，同时还有一条从 v_j 到 v_i 的有向路径，则称两个顶点强连通。如果有向图 G 的每两个顶点都强连通，称 G 是一个强连通图。有向图的极大强连通子图，称为强连通分量。

bzoj1093 ZJOI2007 最大半连通子图

一个有向图 G=(V,E) 称为半连通的，当且仅当对于图中任意两点 u，v，存在一条 u 到 v 的有向路径或者从 v 到 u 的有向路径。给定一个有向图 G，请求出 G 的最大半连通子图的节点数，以及不同的最大半连通子图的数目。

题解

首先可以缩点，变成一个 DAG，那么可以发现一个半连通子图一定是 DAG 上的一条链，那么直接 dp 就可以了。

loj2255 SNOI2017 炸弹

在一条直线上有 N 个炸弹，每个炸弹的坐标是 X_i，爆炸半径是 R_i，当一个炸弹爆炸时，如果另一个炸弹所在位置 X_j 满足：

$$X_i - R_i \le X_j \le X_i + R_i$$

那么，该炸弹也会被引爆。

现在，请你帮忙计算一下，先把第 i 个炸弹引爆，将引爆多少个炸弹呢？

$N \le 10^5$

题解

- 首先，可以求出第 i 个炸弹引爆后直接影响的区域 L_i，R_i。

题解

- 首先，可以求出第 i 个炸弹引爆后直接影响的区域 L_i，R_i。
- 然后将 i 向 L_i 到 R_i 之间的所有点连有向边，这里可以用线段树来优化连边。

题解

- 首先，可以求出第 i 个炸弹引爆后直接影响的区域 L_i，R_i。
- 然后将 i 向 L_i 到 R_i 之间的所有点连有向边，这里可以用线段树来优化连边。
- 那么接下来的问题就是在这个有向图中，计算每个点能够到达多少个点。

题解

- 首先将每个强联通分量缩点，变成一个 DAG，然后在 DAG 上做这个问题。

题解

- 首先将每个强联通分量缩点，变成一个 DAG，然后在 DAG 上做这个问题。
- 在一般情况下这个问题只能 n^2 地做，因为要求集合的并。

题解

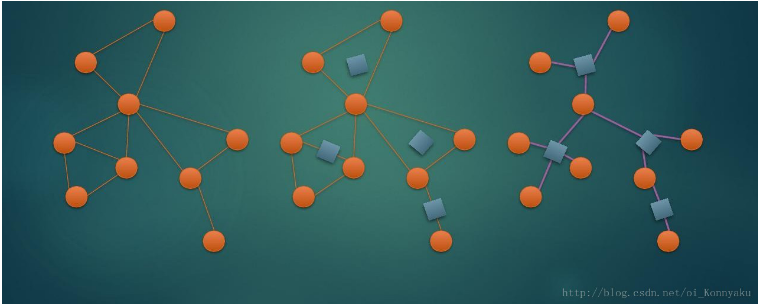
- 首先将每个强联通分量缩点，变成一个 DAG，然后在 DAG 上做这个问题。
- 在一般情况下这个问题只能 n^2 地做，因为要求集合的并。
- 但是这道题有一个特殊性质，就是每个点能够到的点一定是编号连续的一段。

题解

- 首先将每个强联通分量缩点，变成一个 DAG，然后在 DAG 上做这个问题。
- 在一般情况下这个问题只能 n^2 地做，因为要求集合的并。
- 但是这道题有一个特殊性质，就是每个点能够到的点一定是编号连续的一段。
- 那么集合的并就很好求了。

简介

圆方树是很有用的东西。



这样建出来的圆方树圆点之间没有边，方点之间也没有边。

来源不明

给定无向图，每次询问是否存在从 a 到 c 到 b 的点不重复路径，abc 互不相同。

题解

- abc 是合法的当且仅当在圆方树上，c 相邻的点在 a 到 b 的路径上。

题解

- abc 是合法的当且仅当在圆方树上，c 相邻的点在 a 到 b 的路径上。
- 那么直接做就可以了。

题解

- abc 是合法的当且仅当在圆方树上，c 相邻的点在 a 到 b 的路径上。
- 那么直接做就可以了。
- 只需要检查 c 的父亲以及 ab 的 lca。

loj2587 APIO2018 铁人两项

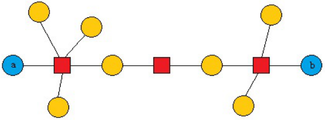
给定无向图，计算三元组 (a,b,c) 满足存在从 a 到 c 到 b 的点不重复路径，abc 互不相同。

题解

- 跟上一题一样，只不过变成了统计合法方案。

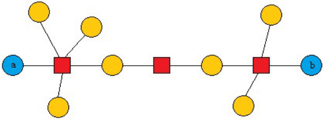
题解

- 跟上一题一样，只不过变成了统计合法方案。
- 假设固定了 a 和 b，那么所有可能的 c 就是下图中的黄色点：



题解

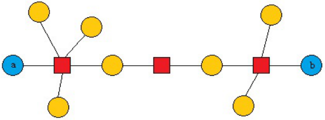
- 跟上一题一样，只不过变成了统计合法方案。
- 假设固定了 a 和 b，那么所有可能的 c 就是下图中的黄色点：



- 可以令每个方点的权值等于其度数，圆点的权值等于 -1，那么黄点的数量就等于 a 到 b 路径上的权值和。

题解

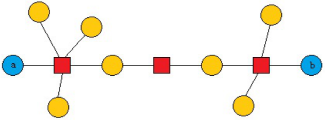
- 跟上一题一样，只不过变成了统计合法方案。
- 假设固定了 a 和 b，那么所有可能的 c 就是下图中的黄色点：



- 可以令每个方点的权值等于其度数，圆点的权值等于 -1，那么黄点的数量就等于 a 到 b 路径上的权值和。
- 于是便可以线性统计答案。

题解

- 跟上一题一样，只不过变成了统计合法方案。
- 假设固定了 a 和 b，那么所有可能的 c 就是下图中的黄色点：



- 可以令每个方点的权值等于其度数，圆点的权值等于 -1，那么黄点的数量就等于 a 到 b 路径上的权值和。
- 于是便可以线性统计答案。
- 这题应该有一个不用圆方树的高妙做法，现场哪位同学来讲一下吗？

并查集中可以使用路径压缩。

状态压缩是将多维状态压缩成一个数字的方法，一般来说最常见的状态压缩是二进制状态。

所以状态压缩本质就是高维 dp。不过这并没有什么用。

loj2318 NOIP2017 宝藏

参与考古挖掘的小明得到了一份藏宝图，藏宝图上标出了 n 个深埋在地下的宝藏屋，也给出了这 n 个宝藏屋之间可供开发的 m 条道路和它们的长度。

小明决心亲自前往挖掘所有宝藏屋中的宝藏。但是，每个宝藏屋距离地面都很远，也就是说，从地面打通一条到某个宝藏屋的道路是很困难的，而开发宝藏屋之间的道路则相对容易很多。

小明的决心感动了考古挖掘的赞助商，赞助商决定免费赞助他打通一条从地面到某个宝藏屋的通道，通往哪个宝藏屋则由小明来决定。

在此基础上，小明还需要考虑如何开凿宝藏屋之间的道路。已经开凿出的道路可以任意通行不消耗代价。每开凿出一条新道路，小明就会与考古队一起挖掘出由该条道路所能到达的宝藏屋的宝藏。另外，小明不想开发无用道路，即两个已经被挖掘过的宝藏屋之间的道路无需再开发。

新开发一条道路的代价是：

这条道路的长度 * 从赞助商帮你打通的宝藏屋到这条道路起点的宝藏屋所经过的宝藏屋的数量（包括赞助商帮你打通的宝藏屋和这条道路起点的宝藏屋）。

请你编写程序为小明选定由赞助商打通的宝藏屋和之后开凿的道路，使得工程总代价最小，并输出这个最小值。

题解

- 发现节点的深度不好处理，那么就按深度 dp。

题解

- 发现节点的深度不好处理，那么就按深度 dp。
- 设 $f[S][i]$ 表示当前选了 S 这些点，树的深度为 i，最小代价是多少。

题解

- 发现节点的深度不好处理，那么就按深度 dp。
- 设 $f[S][i]$ 表示当前选了 S 这些点，树的深度为 i，最小代价是多少。
- 转移的时候，枚举深度为 $i+1$ 的集合 T，然后这次转移的代价就是 $(i+1)*mincost(S,T)$，其中 $mincost(S,T)$ 表示将 T 中的点分别与 S 相连的最小代价。

题解

- 发现节点的深度不好处理，那么就按深度 dp。
- 设 $f[S][i]$ 表示当前选了 S 这些点，树的深度为 i，最小代价是多少。
- 转移的时候，枚举深度为 $i+1$ 的集合 T，然后这次转移的代价就是 $(i+1)*mincost(S,T)$，其中 $mincost(S,T)$ 表示将 T 中的点分别与 S 相连的最小代价。
- 这样可能会得到一些代价比实际大的转移，不过最优解肯定可以被算到。

题解

- 发现节点的深度不好处理，那么就按深度 dp。
- 设 $f[S][i]$ 表示当前选了 S 这些点，树的深度为 i，最小代价是多少。
- 转移的时候，枚举深度为 $i+1$ 的集合 T，然后这次转移的代价就是 $(i+1) * mincost(S,T)$，其中 $mincost(S,T)$ 表示将 T 中的点分别与 S 相连的最小代价。
- 这样可能会得到一些代价比实际大的转移，不过最优解肯定可以被算到。
- 通过一些处理，复杂度是 $3^n \cdot n$。

loj2104 TJOI2015 棋盘

有一个 n 行 m 列的棋盘，棋盘上可以放很多棋子，每个棋子的攻击范围有 3 行 p 列。用一个 $3 \times p$ 的矩阵给出了棋子攻击范围的模板，棋子被默认在模板中的第二行，第 k 列，模板中棋子能攻击到的位置标记为 1，不能攻击到的位置是 0 $(1 \le p \le m, 1 \le k \le p)$。输入数据保证模板中的第二行第 k 列是 1。

在要求棋子互相不能攻击到的前提下，求摆放棋子的方案数。

$1 \le n \le 10^6, 1 \le m \le 6$

题解

- 可以把每一行棋子的摆放位置状压起来。

题解

- 可以把每一行棋子的摆放位置状压起来。
- 这样得到一个 $O(n \cdot 4^m)$ 的做法。

题解

- 可以把每一行棋子的摆放位置状压起来。
- 这样得到一个 $O(n \cdot 4^m)$ 的做法。
- 注意到每行转移是相同的，且状态最多只有 $2^m \leq 64$ 种。

题解

- 可以把每一行棋子的摆放位置状压起来。
- 这样得到一个 $O(n \cdot 4^m)$ 的做法。
- 注意到每行转移是相同的，且状态最多只有 $2^m \leq 64$ 种。
- 那么用矩阵加速就可以了。

题解

- 可以把每一行棋子的摆放位置状压起来。
- 这样得到一个 $O(n \cdot 4^m)$ 的做法。
- 注意到每行转移是相同的，且状态最多只有 $2^m \leq 64$ 种。
- 那么用矩阵加速就可以了。
- 这种题其实挺没意思的。

loj2325 清华集训 2017 小 Y 和恐怖的奴隶主

小 Y 是一个喜欢玩游戏的 OIer。一天，她正在玩一款游戏，要打一个 Boss。

虽然 Boss 有 10^{100} 点生命值，但它只带了一个随从——一个只有 m 点生命值的“恐怖的奴隶主”。

这个“恐怖的奴隶主”有一个特殊的技能：每当它被扣减生命值但没有死亡（死亡即生命值 ≤ 0），且 Boss 的随从数量小于上限 k，便会召唤一个新的具有 m 点生命值的“恐怖的奴隶主”。

现在小 Y 可以进行 n 次攻击，每次攻击时，会从 Boss 以及 Boss 的所有随从中的等概率随机选择一个，并扣减 1 点生命值，她想知道进行 n 次攻击后扣减 Boss 的生命值点数的期望。

$1 \leq n \leq 10^{18}, 1 \leq m \leq 3, 1 \leq k \leq 8$。

题解

- 如果把随从的生命值状压起来，可以发现状态数只有区区 $C_{k+m}^{m} \leq 165$ 种。

题解

- 如果把随从的生命值状压起来，可以发现状态数只有区区 $C_{k+m}^{m} \leq 165$ 种。
- 那么直接套用上一题的方法即可。

题解

- 如果把随从的生命值状压起来，可以发现状态数只有区区 $C_{k+m}^{m} \leq 165$ 种。
- 那么直接套用上一题的方法即可。
- 这题其实有多组数据 $T = 1000$，需要一些其他的处理，不过这个不在这次的讨论范围之内。

bzoj4036 HAOI2015 按位或

刚开始你有一个数字 0，每一秒钟你会随机选择一个 $[0, 2^n - 1]$ 的数字，与你手上的数字进行或操作。选择数字 i 的概率是 p_i。保证 $0 \le p_i \le 1, \sum p_i = 1$。

问期望多少秒后，你手上的数字变成 $2^n - 1$。

$n \le 20$

题解

- 这题就十分小清新了。

题解

- 这题就十分小清新了。
- 先套一个 min-max 容斥将问题转化成对于每个子集，求它期望多少秒后被染到色。

题解

- 这题就十分小清新了。
- 先套一个 min-max 容斥将问题转化成对于每个子集，求它期望多少秒后被染到色。
- 那么接下来只需要一个子集和变换就可以了。

bzoj2734 HNOI2012 集合选数

满足以下条件的正整数集合被称为好集合：若 x 在该集合中，则 2x 和 3x 不能在该集合中。

对于一个正整数 $n \le 10^5$，求出 $1, 2, ..., n$ 的好子集的个数。

题解

- 我们可以构造形如以下的一个矩阵:

$$
\begin{array}{ccccc}
x & 3x & 9x & 27x & \cdots \\
2x & 6x & 18x & 54x & \cdots \\
4x & 12x & 36x & 108x & \cdots \\
8x & 24x & 72x & 216x & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}
$$

题解

- 将 $x=1$ 代入试试：

$$
\begin{array}{ccccc}
1 & 3 & 9 & 27 & \cdots \\
2 & 6 & 18 & 54 & \cdots \\
4 & 12 & 36 & 108 & \cdots \\
8 & 24 & 72 & 216 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}
$$

题解

- 将 $x = 1$ 代入试试：

$$\begin{matrix} 1 & 3 & 9 & 27 & \cdots \\ 2 & 6 & 18 & 54 & \cdots \\ 4 & 12 & 36 & 108 & \cdots \\ 8 & 24 & 72 & 216 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix}$$

- 可以发现，该矩阵有 $log_2 n \leq 17$ 行，$log_3 n \leq 11$ 列。

题解

- 将 $x = 1$ 代入试试：

$$\begin{matrix} 1 & 3 & 9 & 27 & \cdots \\ 2 & 6 & 18 & 54 & \cdots \\ 4 & 12 & 36 & 108 & \cdots \\ 8 & 24 & 72 & 216 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix}$$

- 可以发现，该矩阵有 $log_2 n \leq 17$ 行，$log_3 n \leq 11$ 列。
- 并且只要选的数中没有相邻的元素，那么就是合法的。

题解

- 将 $x = 1$ 代入试试：

$$\begin{matrix} 1 & 3 & 9 & 27 & \cdots \\ 2 & 6 & 18 & 54 & \cdots \\ 4 & 12 & 36 & 108 & \cdots \\ 8 & 24 & 72 & 216 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix}$$

- 可以发现，该矩阵有 $log_2 n \leq 17$ 行，$log_3 n \leq 11$ 列。
- 并且只要选的数中没有相邻的元素，那么就是合法的。
- 那么就可以状压 dp 了。

题解

- 但是注意到这个矩阵中并没有包括所有数，比如 5 就被漏掉了。

题解

- 但是注意到这个矩阵中并没有包括所有数，比如 5 就被漏掉了。
- 没有关系，将 $x = 5$ 代入，又可以构造出一个矩阵，并且肯定是比 $x = 1$ 的矩阵更小。

题解

- 但是注意到这个矩阵中并没有包括所有数，比如 5 就被漏掉了。
- 没有关系，将 $x = 5$ 代入，又可以构造出一个矩阵，并且肯定是比 $x = 1$ 的矩阵更小。
- 由于不同矩阵之间不会发生冲突，所以对于每个矩阵分别计算，最后乘起来就可以了。

bzoj3812 主旋律

给定有向图，询问有多少种删边的方案满足剩余部分强联通。

$n \leq 15$

题解

- 直接算有点难，那么就算不强联通的方案。

题解

- 直接算有点难，那么就算不强联通的方案。
- 若图不强联通，则缩点成 DAG 后会有一些入度为 0 的点，可以通过枚举这些点来计数。

题解

- 直接算有点难，那么就算不强联通的方案。
- 若图不强联通，则缩点成 DAG 后会有一些入度为 0 的点，可以通过枚举这些点来计数。
- 设 $E(S)$ 表示 S 这些点之间边的数量，$C(S,T)$ 表示 S 到 T 之间边的数量，$f(S)$ 表示 S 这些点强联通的方案数。

题解

- 直接算有点难，那么就算不强联通的方案。
- 若图不强联通，则缩点成 DAG 后会有一些入度为 0 的点，可以通过枚举这些点来计数。
- 设 $E(S)$ 表示 S 这些点之间边的数量，$C(S,T)$ 表示 S 到 T 之间边的数量，$f(S)$ 表示 S 这些点强联通的方案数。
- 那么假如枚举缩点后入度为 0 的点集 $T \subseteq S$，方案数就是 $f(T) * 2^{C(T,S-T)} * 2^{E(S-T)}$。

题解

- 然而这是个假算法。

题解

- 然而这是个假算法。
- 因为一个 DAG 中可以有多个入度为 0 的点，那么就会算重。

题解

- 然而这是个假算法。
- 因为一个 DAG 中可以有多个入度为 0 的点，那么就会算重。
- 所以需要用容斥，变成了枚举子集 T 由 k 个强联通分量组成，那么对 $f(S)$ 的贡献就是

$$(-1)^k * 2^{C(T,S-T)} * 2^{E(S-T)}$$

题解

- 然而这是个假算法。
- 因为一个 DAG 中可以有多个入度为 0 的点，那么就会算重。
- 所以需要用容斥，变成了枚举子集 T 由 k 个强联通分量组成，那么对 $f(S)$ 的贡献就是

$$(-1)^k * 2^{C(T,S-T)} * 2^{E(S-T)}$$

- 为了方便计算，设

$$g(S) = \sum_{\text{S 分成 k 个 SCC}} (-1)^k$$

题解

- 那么

$$f(S) = 2^{E(S)} + \sum_{T \subseteq S, T \neq \emptyset} g(T) * 2^{C(T, S-T)} * 2^{E(S-T)}$$

题解

- 那么

$$f(S) = 2^{E(S)} + \sum_{T \subseteq S, T \neq \emptyset} g(T) * 2^{C(T, S-T)} * 2^{E(S-T)}$$

- g 更好算

$$g(S) = -f(S) - \sum_{T \subset S, T \neq \emptyset} f(T) * g(S-T)$$

题解

- 那么

$$f(S) = 2^{E(S)} + \sum_{T \subseteq S, T \neq \emptyset} g(T) * 2^{C(T,S-T)} * 2^{E(S-T)}$$

- g 更好算

$$g(S) = -f(S) - \sum_{T \subset S, T \neq \emptyset} f(T) * g(S-T)$$

- 并且这样完美地避开了计算 $f(S)$ 时枚举到 $T = S$ 时 $g(T)$ 中不能包含 $f(S)$ 的问题。

题解

- 那么

$$f(S) = 2^{E(S)} + \sum_{T \subseteq S, T \neq \emptyset} g(T) * 2^{C(T, S-T)} * 2^{E(S-T)}$$

- g 更好算

$$g(S) = -f(S) - \sum_{T \subset S, T \neq \emptyset} f(T) * g(S-T)$$

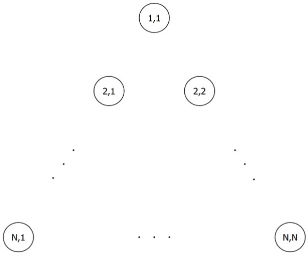
- 并且这样完美地避开了计算 $f(S)$ 时枚举到 $T = S$ 时 $g(T)$ 中不能包含 $f(S)$ 的问题。
- E 和 C 的计算实现精细的话也可以做到 $O(3^n)$。

AGC017F ZigZag

有一个高为 n 的三角点阵。Takahashi 想在上面通过连接点来画 m 条折线，每条折线都从 $(1,1)$ 出发，经过左下角或右下角的点，直到到达第 n 层，折线不能互相穿过，从左到右编号为 1 到 m。

有 k 个限制，每个限制形如 p, i, c。若 $c = 0$，表示第 p 条折线在第 i 层必须往左下角走，若 $c = 1$ 则表示右下角。

Takahashi 想知道有多少种画折线的方法。$n, m \leq 20$



题解

- 一条线可以用状压成一个二进制数，那么可以轻易得到一个 $O(4^n m)$ 的做法。

题解

- 一条线可以用状压成一个二进制数，那么可以轻易得到一个 $O(4^n m)$ 的做法。
- 如何优化，发现每次枚举下一条线十分愚蠢。考虑改成从上往下确定下一条线。

题解

- 一条线可以用状压成一个二进制数，那么可以轻易得到一个 $O(4^n m)$ 的做法。
- 如何优化，发现每次枚举下一条线十分愚蠢。考虑改成从上往下确定下一条线。
- 只需将 dp 状态改为 $f[i][j][S]$ 表示第 i 条线已经确定了前 j 个点，S 的前 $j-1$ 位表示已经确定部分的形状，S 后面的位表示剩余的能走的区域的范围。

题解

- 一条线可以用状压成一个二进制数，那么可以轻易得到一个 $O(4^n m)$ 的做法。
- 如何优化，发现每次枚举下一条线十分愚蠢。考虑改成从上往下确定下一条线。
- 只需将 dp 状态改为 $f[i][j][S]$ 表示第 i 条线已经确定了前 j 个点，S 的前 $j-1$ 位表示已经确定部分的形状，S 后面的位表示剩余的能走的区域的范围。
- 这样复杂度就优化为 $O(nm2^n)$，由于是 atcoder，所以能过。