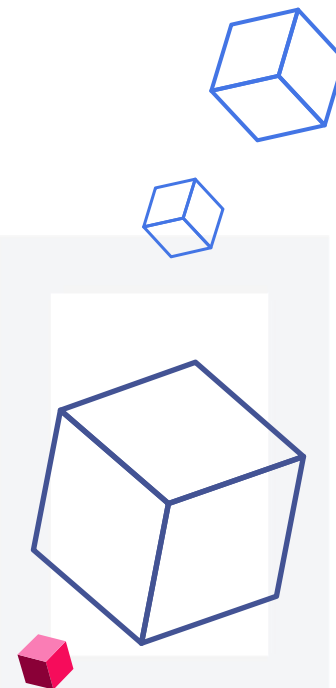


搜索

WX



问题描述:

给定一个 N 个点(编号 $1 \sim N$), M 条边的图。

请你找出若干个环, 满足:

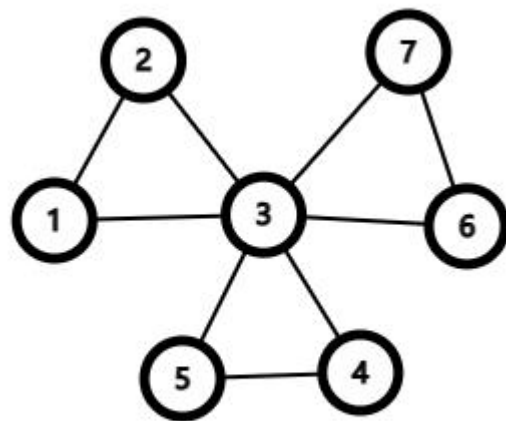
1. 这些环没有重复的边。
2. 这些环覆盖了所有的点和边。

$3 \leq N, M \leq 500000$

问题分析:

- 搜索。
- 需要遍历所有边，以边是否访问为判断条件进行搜索。
- 当一个环上的点遍历完后，肯定会出栈，记录出栈顺序，根据出栈顺序找到不同的环。

问题分析:



- dfs 遍历顺序为: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 3$
- 出栈顺序: 1,3,7,6,3,5,4,3,2,1
- 得到的环有: (3,6,7)、(3,4,5)、(1,2,3)。

问题分析:

- 优化:
- 度为 n 的点 x 会被遍历 n 次, 每次访问更新 $\text{head}[x]$, 指向未访问的边, 避免每次重复遍历访问过的边。
- 时间复杂度约为 $O(M)$ 。
- 来源: [BalticOI2014]Senior Postmen

问题描述:

给定一个 $1 \sim N$ 的排列 P

你需要构造一个括号序列, 满足以下条件:

1. 括号序列是合法的
2. 若第 i 个括号是左括号, 则将 i 和 P_i 连边, 使得最后形成的图满足所有点的度均为 1。

保证有解。

$N=10,40,100$

问题分析:

- 可以发现，排列 P 形成的图是若干个环，且每个环都是偶环(因为保证有解)。
- 对于每一个环，选择了一条边，相邻的边就不能选，因此每个环只有两种情况。
- 枚举每个环选择的情况，找出符合条件的解。
- 考虑最坏的情况，2 个点一个环，有 2^{50} 种情况。
- 考虑优化。

问题分析:

- 如何判断一个括号序列是否合法呢?
- 前缀左括号数量大于等于前缀右括号数量。
- 对于 2 个点一个环的情况，直接将编号小的作为左括号，编号大的作为右括号。
- 这样，只存在 ≥ 4 的环，最多有 2^{25} 种情况，可以直接搜索。
- 时间复杂度为 $O(2^{25})$ 。
- 来源：[雅礼集训2017]蛐蛐国的修墙方案

问题描述:

给定一个 $N \times M$ 的矩形色板，有 K 种不同的颜料，有些格子已经填上了某种颜色。

现在需要将其他格子也填上颜色，使得从左上角到右下角的任意路径经过的格子都不会出现两种及以上相同的颜色。

路径只能沿着相邻的格子，且只能向下或者向右。

计算所有可能的方案，结果对 $10^9 + 7$ 求模。

$1 \leq N, M \leq 1000, K < 20$

问题分析:

- 可以发现, 当 $n+m-1 > k$ 时, 显然无解, 经过的颜色肯定超过了两种及其以上的颜色。
- 合法的 n, m 范围比较小, 可以使用搜索。

问题分析:

- 使用状压的方式记录当前使用的颜色, $f[x][y]$ 表示搜索到 (x,y) 使用的颜色集合。
- 剪枝优化:
 - 可行性剪枝: 剩下的步数大于可用的颜色数, 结束。
 - 排除等效冗余: 如果当前有 k 种颜色还未使用, 对于空白格子染任何一种染色的贡献都是相同的, 只需要选择一种颜色计算, 其他颜色累加即可。

来源: [CF293B]Distinct Paths

问题描述:

给定 D ，求满足 $\text{rev}(N)=N+D$ 的 N 的个数， $\text{rev}(N)$ 表示十进制下将 N 按位翻转并去掉前导 0 后的数。

$$1 \leq D \leq 10^9$$

问题分析:

- 设 N 由 a_0, a_1, \dots, a_k 组成, 则 $N = \sum_{i=0}^k a_i 10^i$
- $d = \text{rev}(N) - N = \sum_{i=0}^k (10^{k-i} - 10^i) a_i$
- 因为翻转后 i 和 $k-i$ 相同, 还可以继续化简:
- $d = \text{rev}(N) - N = \sum_{i=0}^{\frac{k-1}{2}} (10^{k-i} - 10^i) (a_{k-i} - a_i)$
- 例如有 6 为数为 $N=abcdef$, $\text{rev}(N)=fedcba$
- $\text{rev}(N)-N=(10^5-10^0)(f-a)+(10^4-10^1)(e-b)+(10^3-10^2)(d-c)$

问题分析:

- 设 $b_i = a_{k-i} - a_i$, $b_i \in [-9, 9]$
- $d = \sum_{i=0}^{k-1} \frac{2}{10} (10^{k-i} - 10^i) b_i$
- 设 N 的长度为 L , 可以证明 N 的长度不会超过 d 的一倍, 即 $L \leq 18$ 。
- 此时, 就可以直接枚举长度 L 和 b_i , 但时间复杂度无法接受, 继续优化。

问题分析:

- 设 当前枚举到 b_i
- 因为 $10^{k-i} - 10^i > 9 \sum_{j=i+1}^{k-1} (10^{k-j} - 10^j)$
- 枚举完 b_i , 设剩下的为 d' , 此时 d' 必须满足
- $-(10^{k-i} - 10^i) < d' < 10^{k-i} - 10^i$
- 因此, 实际满足情况的 b_i 最多有 2 个, 可以直接枚举。
- 枚举 N 的长度 L 和 b_i , 将每一位 b_i 的方案数相乘就是长度为 L 时的方案数。
- 总时间复杂度为 $O(18 \times 2^9)$ 。
- 来源: [ARC075D]Mirrored

问题描述:

我方有 n 个士兵，敌方有 m 个士兵，每个士兵的血量在 $1 \sim 6$ 之间。

现在有 d 轮攻击，每轮攻击会等概率选择一个活着的士兵，使得血量 -1 ，如果血量为 0 ，则表示死亡。

问，敌方士兵全部死亡的概率是多少？

$1 \leq n, m \leq 5, 1 \leq d \leq 100$

问题分析:

- 因为 n, m 很小，可以直接搜索。
- 如何记录搜索的状态？
- 使用一个十位数来记录每一个士兵当前的血量。
- 但这样的状态太多，有 6^{10} 种，考虑优化。
- 有很多士兵的血量是一样的，可以统一计算。考虑记录每种血量士兵的数量。使用一个十二位数进行记录，前六位记录我方士兵，后六位记录敌方士兵，这样状态会更少。
- 同时进行记忆化，记录搜索过的情况。
- 来源：[GYM101933E]Explosion Exploit

问题描述:

有一个监狱，关押着两个犯人。

监狱可以看做 $h \times w$ 的矩阵，包含四种 类型：

- `.` 表示空地，可以行走。
- `*` 表示墙，无法行走。
- `#` 表示门，初始时时关闭，但可以打开然后行走。
- `S` 表示犯人的位置。

现在，两个犯人要逃出监狱(走出矩阵)，为了避免打草惊蛇，他们需要尽可能的少开门逃走。

问，两个犯人逃出监狱至少要打开多少个门。T组测试数据。

$$2 \leq h, w \leq 100, T \leq 100$$

问题分析:

- 将监狱的外围一圈全部看作空地，任选一个空地位置看作第三个犯人。
- 问题就转换为了三个犯人相聚需要开门的最少数量。
- 对于每个犯人，计算出到每一个位置的需要开门的次数。将门的代价看作 1，空地的代价看作 0，进行 01BFS，使用双端队列实现。
- 最后枚举相聚的位置，更新最优解。
- 时间复杂度为 $O(Thw)$ 。
- 来源：[Gym100625J]Jailbreak

问题描述:

有 n 个任务和三个人 L,M,W。

每次任务给出每个人参与后能得到的分数，每次任务需要两个人完成。

为了保证 n 个任务结束后三个人得到的分数是一样的，且尽量大。请你输出每次任务要派哪两个人，如果不行，输出 impossible。

$n \leq 25$

问题分析:

- 每次任务人员组合有 3 种，直接搜索，时间复杂度为 $O(3^{25})$ 。
- 考虑折半搜索。

问题分析:

- 那么如何合并前后的搜索结果呢?
- 设前半部分搜索三个人的分数为 a, b, c , 后半部分搜索, 三个人的分数为 x, y, z 。
- 则需要满足: $a+x=b+y=c+z$ 。
- 将上式进行转换: $a-c=z-x, a-b=y-x$ 。
- 因此, 搜索前半部分记录 $(a-c, a-b)$, 相同的记录 a 更大的情况。搜索后半部分, 查找 $(z-x, y-x)$ 是否出现过, 出现则更新答案。
- 方案可以使用三进制 0,1,2 进行记录。
- 时间复杂度为 $O(3^{n/2})$
- 来源: [CF585D]Lizard Era: Beginning



T8



西南大学附属中学
High School Affiliated to Southwest University

问题描述:

给定一个长度为 n 的正整数数组 A_i 。

求是否存在 x ，使得对于数组 B_i ，其中 $B_i = A_i \text{ xor } x$ ， B_i 中所有元素两两 $\text{popcount}(B_i)$ 相同。

其中 xor 是按位异或， $\text{popcount}(x)$ 是 x 的二进制表示中 1 的个数。

请输出任意满足条件的 x ，若不存在输出 -1。

$2 \leq n \leq 100, 0 \leq A_i, x \leq 2^{30}-1$

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University

问题分析:

- 折半搜索。
- 将 x, a 的二进制都拆分为前 15 位和后 15 位进行枚举。
- 设 b_i 表示 x 前 15 位与 a_i 前 15 位异或后 1 的数量。
- 设 c_i 表示 x 后 15 位与 a_i 后 15 位异或后 1 的数量。
- 需要满足所有 $b_i + c_i$ 相等, 即 $b_1 + c_1 = b_2 + c_2 = \dots = b_i + c_i$ 。
- 如何判定呢?

问题分析:

- 只考虑 $b_i + c_i = b_1 + c_1$, 将式子转换一下: $b_i - b_1 = c_1 - c_i$ 。
- 求解 b 时, 将每一个 x 对应的 $b_i - b_1$ 都存储下来, 求解 c 时, 将每一个 x 对应的 $c_1 - c_i$ 都存储下来。
- 如果发现某个 x , 每一项对应的 $b_i - b_1$ 和 $c_1 - c_i$ 均相等, 则找到解, 直接输出。
- 时间复杂度为 $O(15n2^{15})$
- 来源: [CF1257F]Make Them Similar

问题描述:

给一个长度为 N 的排列。

有 M 种允许的修改方式，保证修改方式不重复，每种方式用 L, R 来表示，意为你可以将下标为 L 的数与下标为 R 的数交换。你可以修改该排列若干次，请给出一种修改方案，使原排列变为 $1, 2, 3, \dots, N$ 。

如果有多种方案，输出修改次数最少的方案。如果还有多种方案，输出任意一组即可。

$$1 \leq N \leq 12, \quad 1 \leq M \leq N(N-1)/2$$

问题分析:

- IDA*
- 如何设定估价函数?
- 对于一个序列, 每一次操作, 可以改变两个数的位置。如果当前排列与全排列不相同的位置有 k 个, 那么最少操作次数为 $\left\lceil \frac{k}{2} \right\rceil$ 。
- 考虑继续优化:
- 如果当前排列的某个位置和目标位置可以一次操作完成, 则操作次数 +1, 否则操作次数 +2。

问题分析:

- 继续优化:
- 将变换操作看作一个图, 每个位置到达目标位置的最少操作次数就是图上的最短路, 通过 floyd 预处理出来。这样可以求解出当前排列到全排列的最少操作次数。
- 来源: [COCI2009-2010 #2]POSLOZI

问题描述:

有一个 $h \times w$ 的地板。

每个位置有两种情况：1 表示损坏，0 表示正常。

现在你可以用任意大小的正方形木板覆盖损坏的部分，任意大小木板数量充足，且可以重叠。

问至少需要多少块木板才可以覆盖所有损坏的部分。

注意：木板覆盖的部分必须是已经损坏的部分，不能覆盖正常部分。

$$0 \leq h, w \leq 10$$

问题分析:

- 能放大的就尽可能放大的。
- 首先，统计出每个损坏位置作为正方形木板的左下角能放置的最大的正方形，记录最大正方向的边长 mx 。
- 如果一个位置只能被一个正方形覆盖，那么这个正方形必须选，否则加入备选方案。

问题分析:

- 对备选方案的正方形进行搜索，从右下角的位置搜索到左下角。
- 直接搜索会超时，考虑设计估价函数。
- 统计出当前未覆盖点的数量，假设这些点全部放在 $m \times m$ 的正方形中，至少需要的正方形木板数量。
- 来源：[POJ2032]Square Carpets

问题描述:

给定一个 $h \times w$ 的地图，包含墙和空地。

现在已知 n 个人的位置和各自目的地。

每一次操作你可以**移动任意数量的人数**，每个人可以选择停留，或者沿着上下左右移动一步，且需要满足：

1. 同一个位置不能有 multiple 个人
2. 移动过程中，不能出现两个人位置交换的情况。

问，将 n 个人移动到各自的目的地，最少需要多少次操作？

$$4 \leq h, w \leq 16, \quad n \leq 3$$

问题分析:

- 基本思路就是将初始位置通过 BFS 拓展出可行状态，直到到达目标状态。
- 如何记录状态呢？地图长宽最大是 16 。分别用 4 位二进制表示 x,y ，最多三个人，一共 24 位就可以表示三个人的位置信息。
- 一共有 256^3 种状态，状态数比较多，直接 BFS 会超时。
- 考虑设计估价函数。预处理出每个人在每个位置到达目标点的最少操作次数。
- 则当前状态到最终的目标状态还需要的最少操作次数在三人之间取最大值。
- 来源：[POJ 3523]The Morning after Halloween