题目讲评

HJQwQ

2020 年 8 月 16 日

目录

测试点 1

$n = 1, m \leq 10$
显然 n=1 的时候用 1×2 的骨牌尽可能靠左填就行

测试点 2,3

$n \leq 2, m \leq 10^5$
显然 n=2 的时候用 2×1 的骨牌尽可能靠左填就行

测试点 4,5

$n \le 5, m \le 10^5$

说实话我也没想过这部分该怎么做.. 可能有乱搞能拿 50 分

但其实我小小地卡了一下我能想到的几种贪心乱搞.. 乱搞能过的话算你狠

测试点 6,7

$n \le 10, m \le 1000$

这部分或许可以用插头 DP 来做.. 但写起来肯定会很恶心所以我也没细想..

一个结论

答案就是 $\lfloor\frac{(n+1)(m+1)}{6}\rfloor$
惊不惊喜? 刺不刺激?(仿佛梦回 NOIP2017D1T1)

问题转化

原问题是在 $n \times m$ 的棋盘中，放入若干**不相邻**的 1×2 或 2×1 的骨牌

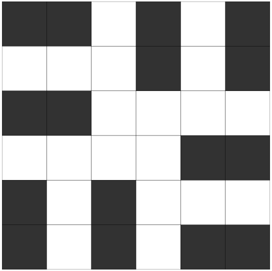
这个问题等价于在 $(n+1) \times (m+1)$ 的棋盘中，放入若干**不重叠**的 2×3 或 3×2 的骨牌

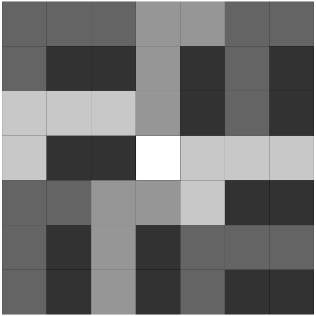
我们不难证明这两个问题中的方案是一一对应的：在第二个问题的任一个方案中，所有的骨牌删去最上方一行和最左侧一列，再删去棋盘的最上方一行和最左侧一列，就能得到第一个问题的一种方案

类似地，在第一个问题的任一个方案中，所有的骨牌在左侧添加一列，在上方添加一行，再在整个棋盘的最左侧添加一列，最上方添加一行，就能得到第二个问题的一种方案

故两个问题中能放置骨牌的最大数目相同，这样我们就完成了对问题的转化

良心插图





原问题中 $n = 6, m = 6$ 的最优方案与新问题中的最优方案可相互转化

$r=2$ 或 $r=3$ 时..

接下来我们讨论的都是转化后的问题
我们将 $(n+1)$ 记作 r, $(m+1)$ 记作 c, 故 $r, c \geq 2$
当 $r=2$ 或 $r=3$ 的时候答案一定是 $\lfloor \frac{rc}{6} \rfloor$, 因为我们只要把骨牌挨着放就行了

$r = 6$ 时

结论: 当 $r = 6$ 时，一定可以放满

首先不难发现当 $r = c = 6$ 时，一定是可以放满的，所以只需要按 $c \bmod 6$ 的值进行讨论

若 $c \bmod 6 = 0$，则直接放满就行

若 $c \bmod 6 = 2$，不难发现 6×2 的很容易放满

若 $c \bmod 6 = 3$，不难发现 6×3 的也很容易放满

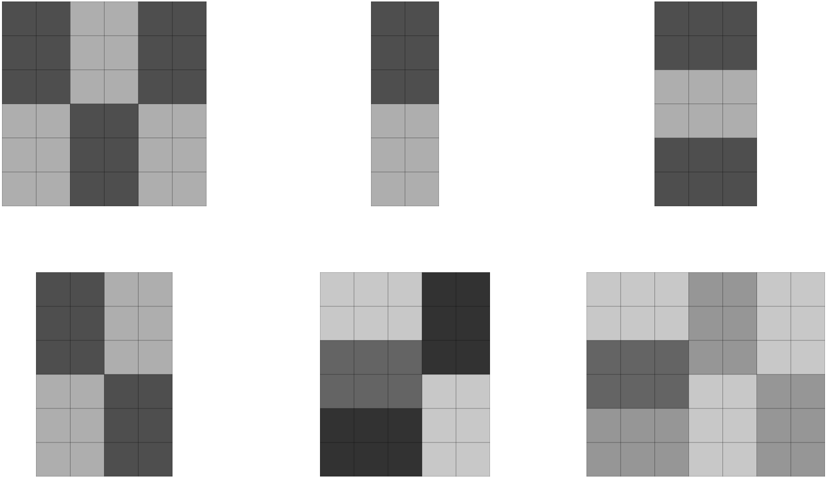
若 $c \bmod 6 = 4$，放两个 6×2 就行

若 $c \bmod 6 = 5$，放一个 6×2 和一个 6×3

若 $c \bmod 6 = 1$，则 $c \geq 7$，放一个 6×3 和两个 6×2 就能填满 6×7

故 $r = 6$ 时一定可以填满,$c = 6$ 时同理

良心插图



六种情况对应的方案

$r = 4$ 时

由于 4×6 的可以填满，所以只需要讨论 $c \bmod 6$ 的值，找到一种填的方法使留空的格子数小于 6

若 $c \bmod 6 = 0$，则直接放满就行

若 $c \bmod 6 = 1$，就不放，反正只留了 4 个空

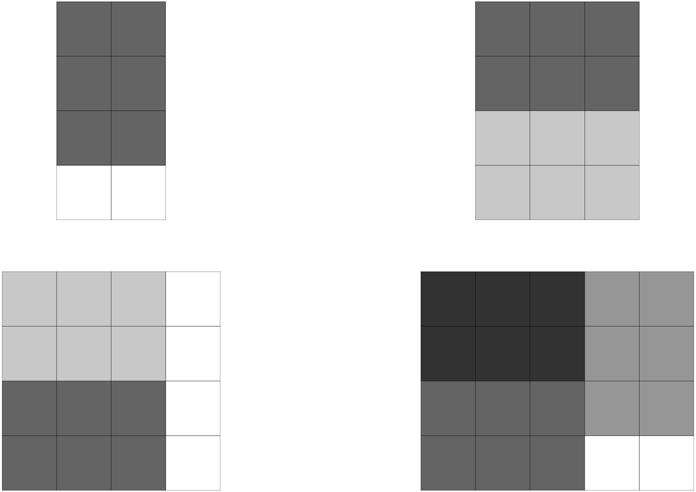
若 $c \bmod 6 = 2$，放一个 3×2

若 $c \bmod 6 = 3$，放两个 2×3

若 $c \bmod 6 = 4$，放两个 2×3

若 $c \bmod 6 = 5$，放两个 2×3 和一个 3×2

良心插图



后四种情况对应的方案

$r = 5$ 时

由于 5×6 的可以填满，所以只需要讨论 $c \bmod 6$ 的值，找到一种填的方法使留空的格子数小于 6

若 $c \bmod 6 = 0$，则直接放满就行

若 $c \bmod 6 = 1$，就不放，反正只留了 5 个空

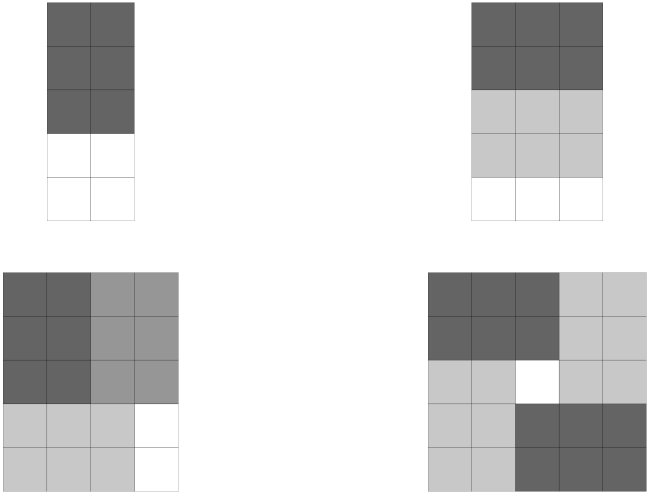
若 $c \bmod 6 = 2$，放一个 3×2

若 $c \bmod 6 = 3$，放两个 2×3

若 $c \bmod 6 = 4$，放两个 2×3 和一个 3×2

若 $c \bmod 6 = 5$，像风车一样放两个 2×3 和两个 3×2，中间留空一格

良心插图



后四种情况对应的方案

$r = 7$ 时

由于 7×6 的可以填满，所以只需要讨论 $c \bmod 6$ 的值，找到一种填的方法使留空的格子数小于 6

若 $c \bmod 6 = 0$，则直接放满就行

若 $c \bmod 6 = 2$，放一个 6×2

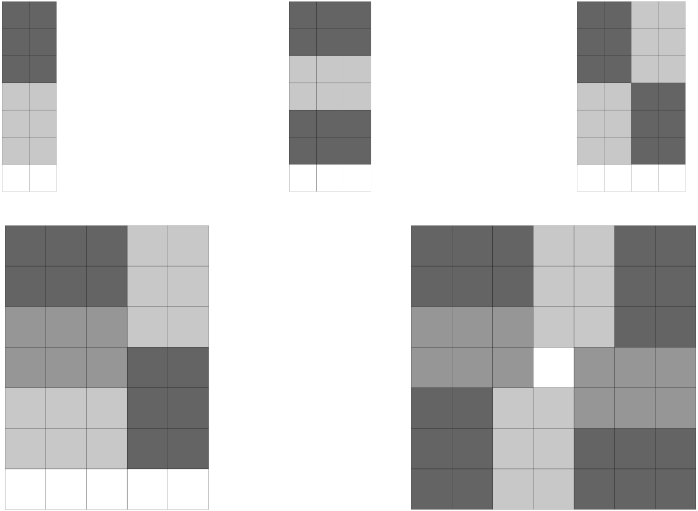
若 $c \bmod 6 = 3$，放一个 6×3

若 $c \bmod 6 = 4$，放一个 6×4

若 $c \bmod 6 = 5$，放一个 6×5

若 $c \bmod 6 = 1$，则 $c \geq 7$，像风车一样放两个 3×4 和两个 4×3 就行

良心插图



后五种情况对应的方案

$r > 7$ 时

由于 r=6 时一定可以填满，我们可以先从棋盘中划去若干列，使 $2 \le c \le 7$，再套用上面的证明，于是这样留的空的数目一定 < 6

综上所述，我们证明了对于所有的情况，都有 $ans = \lfloor \frac{rc}{6} \rfloor$，即 $ans = \lfloor \frac{(n+1)(m+1)}{6} \rfloor$

后记

第一题就讲完惹..

HJQwQ 提醒自己休息一下 qwq

这题本来是去年 NOI 之前某场 CF div2 的 A 题.. 当时是罕见的下午比赛，我就和机房小伙伴开黑.. 结果遇到了这个题

我很快就证明了答案就是 $\lfloor \frac{(n+1)(m+1)}{6} \rfloor$.. 然后一直过不了 pretest.. 自闭了..div2A 都做不出来

然后比赛就被爆破 (unrated) 了.. 出题人和验题人写的是同一种乱搞做法..$n = 6, m = 6$ 时输出 7.. 导致正解过不了 pretest..

然后这题就作为错题从 CF 题库中被删掉了.. 我在 NOI 之前错过了最后一次橙名的机会.. 导致橙名迟到了 1 年 qwq

测试点 1,2

直接暴力 dfs 枚举抽的卡.. 然后统计答案
时间复杂度大概是 $O(n!)$

测试点 3,4,5

动态规划.. 令 $f_{i,j,0/1}$ 表示到达前 i 个位置中选了 j 个, 且第 i 个选/未选的最大价值和

转移方程为:

$$\begin{cases} f_{i,j,0} = max(f_{i-1,j,0}, f_{i-1,j,1}) \\ f_{i,j,1} = f_{i-1,j-1,0} + v_i \end{cases}$$

时间复杂度 $O(n^2)$

测试点 6,7

v_i 不是 1 就是 2.. 大概可以乱搞? 我没仔细想...

一个假贪心

每次抽序列中最大的.. 然后把它以及两侧的牌从序列中删去

这样做满足了不能相邻的要求，但得到的并不是最优解.. 甚至抽不满 $\lfloor \frac{n+1}{2} \rfloor$ 张卡

改进贪心

我们发现，如果 $k=1$ 时最优解为抽 v_i，那么 $k=2$ 时要么抽 v_i 和另一张卡，要么抽 $v_{i-1}+v_{i+1}$

因为如果 k=2 时最优解是抽 $v_{i-1}+v_j$，其中 $j \neq i+1$ 那么说明 $v_j > v_{i+1}$，且 j 与 i 不相邻，又因为 v_i 是原序列的最大值，故抽 v_i+v_j 一定优于 $v_{i-1}+v_j$，矛盾

抽 $v_{i+1}+v_j$ 时同理

于是由 $k=1$ 到 $k=2$ 的过程中，我们要么抽一张和 v_i 不相邻的卡，要么放回 v_i，抽出 v_{i-1} 和 $vi+1$

同理，如果在 $k=2$ 时选了 $v_{i-1}+v_{i+1}$，那么 $k=3$ 时一定再抽一张不相邻的卡，或者抽 $v_{i-2}+v_i+v_{i+2}$

不难发现我们每次都是在将一个类似于 01010 的串翻转为 10101,1 和 0 表示抽或未抽

改进贪心

于是我们就用堆来维护所有翻转可以获得的最大价值 (可能为负)，并用双向链表来维护序列

每次从堆中取出一个点 v_i，就将其加入答案，将 v_{i-1}, v_i, v_{i+1} 从序列中删除，合并为一个新的点，权值为 $v_{i-1} + v_{i+1} - v_i$，再插入到堆和双向链表中

一共要进行 $\lfloor \frac{n+1}{2} \rfloor$ 轮这样的操作

总时间复杂度 $O(n \log n)$

注意特判取出的点 v_i 位于链表两端的情况，以及已经从序列中被删除的情况

证明显然 (x)

emmm 其实挺复杂的.. 直接感性理解就是对贪心进行了改进, 可以对贪心进行反悔操作, 防止陷入局部最优解..

(因为证明太麻烦了所以我本来没打算出这个题.. 但在 GGN 的强烈要求下还是出了.. 他还立了个 flag 说这题没人 AC 他吃点啥 (滑稽))

GGN 巨神给出了一个证明, 课后会发给大家

HJQwQ 表示这题本质上是一个**模拟费用流问题**, 也给出了一个证明..

与费用流的联系

前置知识: 网络流, 费用流,MCMF 算法
这道题可以抽象成一个费用流问题
建图方法如下..
(此处有一张图片)
中间的每一条边流量上限为 1, 费用为 v_i, 代表原来的一张卡, 有流量就代表要选它
左右的边流量上限为 1, 费用为 0, 保证了相邻两张卡只会选一张
那么这显然就是一个最大费用流问题

与费用流的联系

每次 $k++$ 的过程就是费用流的增广过程: 在残量网络中找一条总费用最大的从 s 到 t 的路径, 然后将这条路径的流量增加 1

由于总费用最大的增广路一定是从 s 到第 1 层节点, 然后经过了若干条中间的边, 最后进入 t, 这就等价于找一条翻转价值最大的 01 串

然后将这条路径上每条边的流量 $+1$, 由于流量上限也是 1, 故这样会将残量网络中路径上的边反向, 等价于将这个 01 串翻转

这样就证明了题解的做法等价于费用流 MCMF 算法的增广过程, 由于贪心沿着最长费用路增广是正确的, 故题解中的也是正确的

我们称这类用堆等数据结构来模拟费用流的增广/退流等操作的问题为**模拟费用流问题**

拓展

什么？为什么 MCMF 算法每次沿着最小费用路增广是正确的？

时间原因这里就不再赘述.. 有兴趣的同学可以看巨佬的洛谷题解 大概思路就是先证明消圈定理：当前流是最小费用流等价于图中无负环

然后证明若图中一开始不存在负环，那么沿着最小费用路增广后也一定不会出现负环

后记

第二题也讲完惹..

HJQwQ 提醒自己休息一下 qwq

这道题原题是洛谷 P1484

这题是我在广州集训的时候遇到的.. 当时还不会做.. 然后讲完之后觉得自己就是个锑..

并且这题在很多 OJ 上都出现过很多次..BZOJ 和洛谷至少各有 3 道一样的题..

我能找到的最早出处大概是APIO2007 数据备份

后来就被无数集训/模拟赛/OJ 抄了无数次.. 但极难找到这道题的正确性证明.. 我和 **GGN 巨神**证了好久才证出来

总而言之是一个很经典也很有意思的题 qwq

STL 的优先队列不开 O2 慢的要死..n 只能出到 2×10^5.. 不然本来还想出到 10^6..

测试点 1,2

可以枚举删边，然后 $O(n!)$ 枚举所有可能的路径，再计算经过这条路径的概率和总长度，最后算出期望值

测试点 3,4,5

令 g_i 表示从 i 号点开始走经过的路径长度期望值，则有

$$g_i = \begin{cases} 0, & \text{i 无出边} \\ \sum\limits_{(i,v)\in E} \frac{w_{(i,v)}}{\sum\limits_{(i,u)\in E} w_{(i,u)}}\left(g_v + l_{(u,v)}\right), & \text{i 有出边} \end{cases}$$

这样就可以枚举删边，然后每次用拓扑排序暴力 DP，时间复杂度 $O(m^2)$

测试点 6,7

不太清楚有什么乱搞做法 qwq..

正解

首先通过一次拓扑排序计算出 g_i

然后考虑删边对答案的影响: 删去边 (u,v) 只会对经过 u 点的路径造成影响, 并且只会对 u 点的 g 值造成影响, 故要重新计算 g_u, 并求出 g_u 的变化对总答案的响, 故我们要求出经过 u 点的概率 f_u 令 f_i 表示从 1 号点出发, 前进路径经过 i 号点的概率, 则有

$$f_i = \begin{cases} 1, & i = 1 \\ \sum\limits_{(u,i)\in E} \dfrac{w_{(u,i)}}{\sum\limits_{(u,v)\in E} w_{(u,v)}} f_u, & i \neq 1 \end{cases}$$

正解

于是我们再用一次拓扑排序算出 f_i, 对一条边 (u, v), 计算删去它后,g_u 的变化量 Δg_u, 总答案 ans 的变化量 $\Delta ans = \Delta g_u \times f_u$, 这样算一次是 $O(1)$ 的

然后我们枚举每一条边进行计算, 总答案 ans 取最小值即可, 总时间复杂度 $O(m)$

后记

终于讲完了 qwq

这道题原题是BZOJ3470

原题要保留 6 位小数... 我当时好像卡了好长时间的精度才过

这个题也是综合了概率和期望的很妙的题..

本来想出另外一道题的.. 但因为太容易搜到题解所以被我毙掉了

然后正好想到这几天的模拟赛还没出概率期望.. 就想出一道概率期望

本来打算出一个无向图随机游走的题.. 但 **GGN 巨神**说那种题太水了..HJQwQ 就只好另外找了这道题.. 结果还是被 **GGN 巨神**秒了 qwq

emmm 还有就是 double 运算不开 O2 好慢... 本来打算出到 $m \leq 2 \times 10^6$ 的.. 但试了一下直接就 TLE 了 qwq

完结

祝大家身体健康，再见！