收拾房间 reorder

方法一：

按字典序从小到大枚举 $1\sim n$ 的全排列，然后判断最开始就在书架上的 m 本书，在当前枚举的排列 A 中是否保持原有的顺序，如果是，显然存在一种插入剩余 $n-m$ 本书的方案，得到当前枚举的排列 A，且易知若排列 A 是第一个满足条件的排列，则 A 必定是字典序最小的。时间复杂度 $O(n!)$，期望得分 30 分。

方法二：

贪心。方法一中的枚举显然是很浪费时间的：我们想要找一个最小的排列，使得最开始 m 本书在该排列中保持原有的顺序。那我们本来就不需要枚举这 m 本书的排列顺序。于是可以先把这 m 本书从 $1\sim n$ 的排列中"抽"出来，只考虑剩下的 $n-m$ 本书的排列。

而剩下 $n-m$ 本书的排列中，显然从小到大排序后的编号递增的排列是最小的，我们只需要考虑如何把刚"抽"出来的 m 本书依次插回去，使得最终排列尽可能小。

设当前即将插入的书本编号为 x，当前可以插入的位置$A_i,A_{i+1},\ldots,A_j$ 之间，且存在 k 使得 $A_k<x<A_{k+1}(i-1\le k\le j)$，则书本 x 插入到 A_k 和 A_{k+1} 之间最优。证明略。

若具体实现方法和插入排序一样需要移动数组，则时间复杂度为 $O(n^2)$，期望得分为 50 分；若使用链表或指针等优化后的实现方法，时间复杂度为 $O(n)$，期望得分 100 分。

天天背单词 prefix

题解：

这是一道类似数位 DP 的数学统计题。首先题目保证不存在某个单词是另一个单词的前缀，这既保证了 n 个单词不重复，又保证了任意 k 个单词连成的字符串互不相同（若相同，则该字符串存在至少两种方案分解成 k 个单词，可反证存在某个单词是另一个单词的前缀）。

我们可以先对 n 个单词从小到大排序，依次编号为 1 至 n。然后把字符串 S 分解为 k 个单词并转化为 k 个编号所对应的序列，设其为 $A_1,A_2,\cdots,A_k$。同理，对于任一小于 S 的字符串 T，同样可以得到对应的序列 $B_1,B_2,\cdots,B_k$，且必定存在 i，使得$A_1=B_1,A_2=B_2,\ldots,A_{i-1}=B_{i-1}$ 且 $B_i<A_i$。需要知道 S 在所有合法字符串里排名多少，只需要统计有多少个满足条件的序列 B 即可，因为序列 B 与字符串 T 一一对应。

而要统计满足条件的序列 B 的个数，我们可以按上述的 i 对 B 进行分类，即第 i 类序列 B 满足 $A_1=B_1,A_2=B_2,\ldots,A_{i-1}=B_{i-1}$ 且 $B_i<A_i$。至于计算第 i 类序列 B 的个数，只需要知道对应的 B_i 有多少种选择（剔除 $A_1,A_2,\ldots,A_{i-1}$ 后比 A_i 小的编号个数），再乘以$B_{i+1},\ldots,B_k$ 的组合方案数 $((n-i)!/(n-k)!)$，应用乘法原理两数相乘即可。

代码实现时注意取模运算相关技巧即可。时间复杂度 $O(L\log L)$，期望得分 100 分。

养盆栽 monotony

方法一：

对于其中 65 的数据，直接枚举移动方案然后逐行逐列判断单调性就好，时间复杂度 $O(r*c*2^{(r+c)})$，期望得分 65 分。

方法二：

显然像方法一那样同时枚举行列的选择方案是会超时的，但其实这样枚举很耗费时间，而且有很多枚举是冗余的。想要减少冗余，我们可以首先在枚举了行的选择方案 S 以后，剔除掉一些不满足单调性的列。假设剔除不满足单调性的列以后，剩下 t 列分别为 $C_1, C_2, \ldots, C_t$，那么容易推出行选择方案为 S 且只选择一列的方案数为 t，下面只考虑至少选择两列的方案。

在至少选择两列的情况下，每一行是递增还是递减是确定的，我们可以用一个二进制数 s 来表示（s 的第 i 位为 1 表示第 i 行递增，否则表示递减或没有选择第 i 行），令数组 $dp_{s,k}$ 表示前 k 列中，选择了至少两列，且最后一列为第 k 列，递增递减状态为 s 的方案数，那么有：

$$dp[s][k] = \sum(dp[s][j] + 1)$$

其中从列 j 到列 k 的递增递减状态恰好为 s。因为 s 是 S 的子集，所以枚举 S 后再枚举 s 的总枚举量为 $O(3^n)$，DP 时间复杂度为$O(m^2)$，总时间复杂度为 $O(3^n * m^2)$。

这样的时间复杂度还是太高，需要进一步优化。我们容易发现，其实状态 s 远没有那么多（确定 S 后，最多只有 $m*(m-1)/2$ 个不同的状态 s），而对于每一个 k，有效的状态 s 不超过 m 个，且这 m 个有效状态与枚举的 m 个状态转移（也就是 j）是一一对应的，所以优化后的总时间复杂度为 $O(2^n * m^2)$。期望得分 100 分。