

# CCF NOIP 2022 解题报告

## “种花” 解题报告

清华大学 迟凯文

### 【概述】

本题以“种花”这一现实场景为背景，主要考查内容为对循环、二维数组的使用，以及前缀和的技巧。本题需要选手能够熟练使用循环和二维数组，并通过计算前缀和的方式统计答案。总体而言难度中规中矩，但多组数据可能导致每一组数据计算答案前未清空数组的选手丢失一定分数。

### 【题目大意】

在一个  $n \times m$  的网格上有些位置可以种花，有些位置不能。

问：有多少种花方案，其形状为 C 或者 F(详见题面)。

每个测试点含  $T$  组数据。

$n, m \leq 1000, T \leq 5$ 。

### 【解法一】

根据题意(详见题面中记号)，枚举  $x_1, x_2, x_3, y_1, y_2, y_3$ ，随后进行暴力的检查即可。

### 【解法二】

在解法 1 中检查时可以使用前缀和。

### 【解法三】

以 C 为例，对每个位置分别求其最大能向右延伸多少步；基于此再统计有多少种先向上延伸再向右延伸的方案；二者乘积即统计了以该位置为左下角的 C 形。

### 【总结】

本题的思维难度较低，对实现细节有一定要求，包括需要选手考虑到多组数据应清空数组以免上一组数据的数值影响本组数据的计算。对比往年 NOIP 的第一题，除多测未清空的情况外，选手对本题的难度评价不一，总体与往年难度等同。

值得注意的是，本题在命题之初没有引入多组数据的问题。但本题随机数据容易使部分考虑不周的算法通过，因此希望能对每个部分设计多个测试点以加强题目数据强度。由于评测系统暂时不支持子任务设计，因此采取了多组数据的方法。建议考虑在评测系统增加这一特性。

## “喵了个喵” 解题报告

清华大学 吴作同

### 【概述】

本题以近期火热的某款消除游戏为题目背景，主要考查分类讨论与构造。本题需要选手掌握一定的抓取问题关键性质的能力以及全面考虑数据形态的能力。考虑到直接想出整个算法难度较高，部分分设计有意引导选手从一些特殊情况开始，逐渐扩展到一般情况下问题的解法，由较少的情況数到较多的情况数逐步迈进。总体而言难度较高。

### 【题目大意】

给定一个长为  $m$  的序列，和  $n$  个空栈。你可以使用以下两种操作来消去序列中的所有数。

- 将序列的开头一个元素移动到一个栈中。如果这个被移动的元素与原来在栈顶的元素相同，则消去这两个元素。
- 选择两个不同的栈。如果这两个栈的栈底元素相同，则消去这两个元素。

保证最多有  $k$  种不同的元素且所有元素的出现次数为偶数。

要求输出一种操作方案，输入保证有解。

$T$  组数据， $m$  的和不超过 2 000 000。

$T \leq 1005$ ,  $n \leq 300$ ,  $k \leq 2n-1$ 。

### 【解法一】

$k = 2n-2$ 。

因为序列只能从开头取出，并且元素只能两两消掉，所以如果一个前缀中每种元素都出现了奇数次，那么当把这个前缀的所有元素都入栈时，栈里至少会剩下每种元素各一个。根据  $k$  与  $n$  的数量关系，考虑留一个空栈，并在剩下的所有栈中放不超过两个元素。这样如果栈中已有和下一个入栈的元素相等的元素，那么可以在下一步从栈顶或栈底直接把这下一个元素和栈里的这个元素消去。若栈中没有和下一个元素相等的元素，那么就意味着前  $n-1$  个栈中至少有一个栈里的元素个数小于 2，于是把这个新元素放在这个栈里即可。代码实现上，可以把 1 号和 2 号元素分配到 1 号栈，依此类推，把  $2i-1$  号和  $2i$  号元素分配到  $i$  号栈。然后维护栈中是否有这个元素以及它的位置。

期望得分 15 分。

### 【解法二】

$n=2$ 。

第二个操作相当于能够将两个栈的栈底接通，相邻的元素也能消去。所以  $n=2$  的问题相



当于在一个双端队列中操作。 $k=2n-1=3$ , 显然当双端队列中只有不超过两个元素时, 顺序是无关紧要的, 且可以使用解法一的策略来消除。但是如果需要加入第三种元素, 夹在中间的那个会在此刻无法直接消去。这时可以再往后看一个元素, 避免这个元素被夹到中间。假设双端队列中的元素是 1 和 2, 然后下一个进来的是 3, 那么可以查看 3 后面的元素, 若为 1, 则可以在 2 的一端插入 3, 这样 1 就不会被夹到中间, 且可以从这一端使用第一种操作将队列里的 1 与即将进队的 1 消除; 若为 3, 则可以在任意一端插入前面的那个 3, 因为无论插入到哪边, 接下来可以在同一个方向插入第二个 3 以消去这两个 3。其他情况可以根据编号的对称性或队列两端的对称性归约到上述两种情况中的一种。

期望得分 15 分, 结合解法一可以得到 30 分。

此外, 我们可以注意到, 此时并不能在线地解决这个问题, 即只根据下一个元素的值立即决定它需要从哪个栈放入。因为之后的序列可以立即给出一个被夹在中间的元素且这种元素此后不再出现。这也意味着, 存在一些在加入一个元素时必须查看序列后面内容的情况。这对思考算法的可行性有一定的帮助。

### 【解法三】

$T=3, n=3, m \leq 14$ 。

这是留给搜索的部分。可以枚举每个元素放进哪个栈。若有两个栈的栈底相同可以立即将这两个元素消去。

期望得分 20 分, 结合前面的部分可以得到 50 分。

### 【解法四】

$n=3$ 。

至此可以发现, 本题的关键之处在于当  $n-1$  个栈中均有两个元素且剩下一个栈为空时, 发现下一个元素和栈里的  $2n-2$  个元素均不相同时, 需要怎么操作。并且我们知道需要查看序列中这个新元素之后的内容来考虑这步操作。

假设一个栈自顶至底是 1, 2, 第二个栈自顶至底是 3, 4, 第三个栈是空的, 下一个元素是 5。如果这之后的元素还是 5, 那就可以直接将这两个元素消掉; 如果这之后的元素是 2 或 4, 那么可以将 5 放在对应栈的栈顶, 然后利用空栈从栈底消掉这个 2 或 4, 这样之后仍有一个空栈且其他栈中的元素个数仍不超过 2。

但如果接下来的元素是 1 或 3, 此时的情况就变复杂了。加入了 5 和这个元素之后, 并不能将这些栈还原为之前的状态。由于这两种情况是对称的, 所以可以假设接下来的元素是 1。接着往后考虑, 如果下一个元素是 2, 那么可以把 5 放进空栈里, 然后从栈顶删去 1 和 2, 这样又有了一个空栈; 如果下一个元素是 4, 可以把 5 叠在 3 的上面, 消完 1 之后再利用空栈把 4 消掉; 如果下一个是 5, 那么可以直接把两个 5 放在空栈里消掉。但如果是 1 或 3 呢?



这时应该已经发现了，我们能确定的时候在于出现了另一个新元素或者一个栈底元素。假设把新元素放进空栈里，那么可以任意地从栈顶消除和在原位放入接下来出现的栈顶元素。如果遇到了另一个新元素，直接在原空栈里消去即可；如果遇到了一个栈底元素，发现这个栈里只有一个元素，那么可以从栈顶消去这个元素，这样又有一个空栈了；如果遇到了一个栈底元素，但这个栈里有两个元素，那么可以考虑像前面的方法一样，把新元素改成放进这个栈里，可这样这个栈的栈顶元素就没法在这个栈里消除了。不过我们注意到，因为前面假设把新元素放进空栈里的时候，遇到这个栈顶元素时栈里有两个元素，所以在这个新元素与这个栈底元素之间，有偶数个原来在这个栈的栈顶元素，于是我们可以让它们在那个空栈里两两消去，然后利用空栈从栈底消去那个栈底元素。这样问题就解决了，且可以直接扩展到任意的  $n$ 。

期望得分 20~50 分。结合前面的部分可以得到 70~100 分。

这一档部分除了用于提示思路外，还有两个作用：一是让使用时间复杂度为  $O(nm)$  的算法稳稳通过；二是为了留给思考状态压缩动态规划或者巧妙搜索的选手一个机会。可以发现，这个做法的过程中所有栈中的元素个数始终不超过 3，所以一个优秀的状态压缩的动态规划或搜索是有可能通过这一档部分分的。

### 【解法五】

正解中为了将时间复杂度优化到  $O(n+m)$ ，需要做到以下几点：

1. 直接维护每个栈里的元素，以及每种元素是否出现在栈里、出现在哪个栈里，以便查询一个元素是否出现、是否是栈底元素；
2. 用一个队列维护当前元素个数小于 2 的栈里的空位，以便加入新元素时能够给这个元素分配一个空位；
3. 注意到向后查找找到另一个新元素或一个栈底元素后，找过的一整段元素的操作可以直接确定，而不需要从这段元素中的某个位置开始再次进行这样的查找，所以暴力做这样查找的复杂度是对的。

### 【总结】

本题的思维难度较高，主要体现在对问题涉及的各种情况的层层剖析。为了减缓这个剖析过程的难度，笔者按照深入这个问题的过程设置了提示性的部分分。值得注意的是，赛后选手对本题的评价难度明显高于命题组赛前对本题的预期难度，主要原因可能是顺着这个部分分思考的过程也许并没有命题组想象的那样自然。

总体来说，本题在 NOIP 的所有题目中难度较高，作为 NOIP 的第二题大约是因为命题组低估了本题的思维难度。也许交换第二题与第三题会更有利于选手的参赛体验。



## “建造军营” 解题报告

清华大学 张艺缤

### 【概述】

本题是一道与图论相关的动态规划题目，以两军交战时“建造军营、看守道路”为背景，涉及 tarjan 算法、树形 dp 等算法和技巧，考查了选手对图论相关经典算法的掌握和运用能力，以及动态规划的分析与状态设计能力，在 NOIP 系列赛事中大致为中等难度。

### 【题目大意】

给定一张  $n$  个点  $m$  条边的无向简单连通图，问有多少种选择一个非空点集和一个边集的方法，使任意删除一条不在被选边集内的边后，被选点集中的所有点仍然保持连通。

数据范围： $n \leq 5 \times 10^5$ ， $m \leq 10^6$ 。

### 【解法一】

暴力枚举所有的选择点集和边集的方案，并加以直接判断（即直接枚举不在边集外的边，用 dfs 或 bfs 检查删除后点集是否连通）。

期望得分 15 分。

### 【解法二】

暴力枚举点集，然后容易发现，只需要分别枚举每条边，判断删除这条边后点集是否仍然连通即可。设这样的边数为  $k$ ，则这个点集就有  $2^k$  种选择边集的方案。

期望得分 35 分。

### 【解法三】

考虑  $m=n-1$  的情况，也就是整张图是一棵树。

观察树上问题的性质：如果选择了某两个点，那么它们之间的所有边都要被选择。

可以使用树形 dp 的方法来解决，但设计状态和转移时应当格外小心，避免出现重复或缺漏计数。一种可行的状态设计如下。

设  $f(i)$  表示：以  $i$  为根的子树内至少选择一个点，且被选择的点必须通过被选择的边与点  $i$  连通的方案数； $g(i)$  表示：以  $i$  为根的子树内一个点也不选，边可以任意选的方案数。

先考虑转移：对于叶结点， $f(i) = g(i) = 1$ ；对于  $i$  的子结点  $j$ ，从原来的  $f(i)$ ， $g(i)$ ， $f(j)$ ， $g(j)$  转移到新的  $f'(i)$ ， $g'(i)$ ，有：

$$f'(i) = f(i)(f(j) + g(j) * 2) + g(i) * f(j)$$

$$g'(i) = g(i) * 2g(j)$$

然后再考虑如何统计答案，这里同样需要格外小心：一种简单的想法是枚举被选点集的 LCA，设其为  $x$ ，但  $f(x)$  并不是我们想要的答案，因为设计状态时没有要求被选的点必须来自  $x$



的至少两棵不同子树(或  $x$  本身), 这使得被选点的 LCA 一定在  $x$  的子树内, 但不一定是  $x$  本身。

解决方案一是增加一个状态  $h(x)$  来记录“被选的点来自  $x$  的同一个子树内”的情况, 这里不再赘述。解决方案二是更改统计答案时枚举的  $x$  的含义: 注意到假设被选点集的 LCA 是点  $y$ , 那么这些点一定通过被选边和  $y$  相连, 而  $y$  再向上还可能继续通过被选边相连, 直到第一次断开为止。这个“第一次断开”的位置是唯一的, 设其为  $x$ , 那么  $x$  连向父亲的边一定不被选, 而  $x$  子树内部的方案数恰好就是我们 dp 算出来的  $f(x)$ 。

但除了  $x$  与父亲的边一定不被选,  $x$  子树外的其他边都是可以任选的, 设这样的边条数为  $p(x)$ , 则当  $x$  为根时,  $p(x)=0$ , 否则  $p(x)=n-\text{size}(x)-1$ , 其中  $\text{size}(x)$  是  $x$  的子树大小。

那么最终的答案即为:  $\text{ans} = \sum_{x=1}^n f(x) \times 2^{p(x)}$ 。

时间复杂度  $O(n)$ , 结合解法二可以获得 60 分。

#### 【解法四】

由解法二启发正解的思路: 容易注意到, 如果一条边删除后整张图仍然保持连通, 那么无论选择什么点集, 这条边都是可选可不选的; 反正, 如果一条边被删除后会使图不连通, 这才是我们需要根据选择的点集不同来判断它是否必须要选的。这样的边在图论中被称为“割边”或“桥”, 整张图会被割边分隔成若干个部分, 每个部分满足删去其中任意一条边后整个部分仍然保持连通, 则称每个部分为一个“边双”。使用 tarjan 算法可以求出所有的“割边”和“边双”。

进一步分析题目性质, 可以发现: 位于同一个“边双”内的点, 无论是选一个还是选多个, 对于边的可能选法都是不影响的。因此可以将每个“边双”缩成一个点, 这些点之间由那些割边相连, “选择”这个点代表选择原来边双中的至少一个点(若原先边双中有  $k$  个点, 这实际上有  $2^k-1$  种选法, 在 dp 时要注意)。

容易证明, 上述缩点后形成的是一棵树(因为缩点后图显然仍连通, 且一定没有环, 因为环上的边一定不是“割边”), 可以使用解法三中的树形 dp 来解决树上问题。最后将求得的答案乘上 2 的(非“割边”条数)次幂即可。

时间复杂度  $O(n+m)$ , 期望得分 100 分。

#### 【总结】

本题是一道综合了图论算法和动态规划的题目, 首先需要选手分析题目性质, 正确理解“删除一条边使得点集不连通”和图论概念“割边”之间的关联, 从而使用 tarjan 算法将问题转化为树上问题; 进一步, 还需要选手对树形 dp 有较好的理解与掌握, 因为本题的 dp 状态设计和最终答案的统计有一定的思维难度, 不仔细思考容易造成重复或遗漏统计。总体而言, 本题难度在 NOIP 中属于中等, 具有一等奖水平的选手应当能通过此题或至少获得较高的部分分。但考虑到本场比赛的其他题目风格和难度, 将本题放在第二题的位置上可能会使选手的得分情况更好。



## “比赛” 解题报告

北京大学 周雨扬

### 【概述】

本题以一个简洁明了的求和公式为出发点, 主要考查内容为线段树。本题需要选手分析线段树的状态设置方法, 通过设置合理的状态与懒标记来简化维护与求解过程。盲目设置状态极有可能导致代码维护困难和编写困难。部分鼓励选手自行探索合理的状态与懒标记设置方式, 同时通过随机排列的设置考查选手面对绝大多数常见情况时候的应对方法。总体而言本题题目难度中等偏上。

### 【题目大意】

给定长度为  $n$  的排列  $a[1] \cdots a[n]$  和  $b[1] \cdots b[n]$ ,  $Q$  次询问, 第  $i$  次询问给定  $1 \leq l \leq r \leq n$ , 你需要返回  $\sum_{p=l}^r \sum_{q=p}^r (\max_{p \leq i \leq q} a[i]) \times (\max_{p \leq j \leq q} b[j])$  的值。答案对  $2^{64}$  取模。  
 $n, Q \leq 250\,000$ 。

### 【解法一】

$n, Q \leq 30$ 。

对于每一次询问, 我们都可以直接根据题目意思直接模拟求和过程。

时间复杂度为  $O(n^3Q)$ , 期望得分 8 分。

### 【解法二】

$n, Q \leq 3000$ 。

首先我们总可以预处理出来排列  $A, B$  的所有区间最大值。对于固定的  $p, q$ , 我们只需要将两个最大值简单相乘即可。

对于每一次询问, 我们不难发现其本质上是对  $p, q$ , 以及其对应的乘积进行了一次二重枚举求和。因此我们也可以通过预处理前缀和的方式加速询问。

时间复杂度为  $O(n^2+Q)$ , 期望得分 20 分。

### 【解法三】

$a[1] \cdots a[n], b[1] \cdots b[n]$  均为均匀随机生成的排列。

对于固定的  $q$ , 虽然可能有很多个  $p$ , 但是不同的区间最大值对不会很多。在排列随机的情况下后缀最小值个数和前缀最小值个数的期望都是  $O(\log n)$  级别的数字, 即使做归并后也最多只有  $O(\log n)$  段。

对于右端点  $q$  做扫描线, 并利用单调队列快速维护对于不同的左端点  $p$ ,  $\max_{p \leq i \leq q} a[i]$  和  $\max_{p \leq j \leq q} b[j]$  的值, 则归并后我们可以得到期望  $O(\log n)$  段不同的乘积。



此时直接对这  $O(\log n)$  个区间做区间加法即可, 而询问变成了区间求和问题, 利用线段树/树状数组可以简单求解。

上述算法期望时间复杂度  $O(n \log^2 n + Q \log n)$ , 在输入随机的情况下足以通过。期望得分 52 分。

#### 【解法四】

$a[1] \cdots a[n]$  均为均匀随机生成的排列。

考虑扩展解法 3, 让其尝试找到  $O(n \log n)$  个矩形, 使得每个矩形对应的所有区间  $[p, q]$ , 其  $\max_{p \leq i \leq q} a[i]$  和  $\max_{p \leq j \leq q} b[j]$  的值都相同。

对于任意  $x$ , 我们维护  $l[x], r[x]$ , 其分别代表了数组  $a$  中下标在  $x$  之前/之后的第一个值大于  $a[x]$  的数字。则对于任意  $l[x] < p \leq x \leq q < r[x]$ ,  $\max_{p \leq i \leq q} a[i] = a[x]$ 。同时对于上述区间  $[p, q]$ , 其显而易见最多只有  $r[x] - l[x] - 1$  种不同的  $\max_{p \leq j \leq q} b[j]$ , 且对于每一种不同的取值, 其包含的区间都可以看成是一个二维平面上的矩形。

此时我们将问题转化为了  $O(n \log n)$  次矩形加和  $Q$  次矩形求和问题。依然可以利用线段树/树状数组解决。期望时间复杂度  $O(n \log^2 n + Q \log n)$ , 期望得分 68 分。

#### 【解法五】

$Q \leq 5$ 。

如果只有单次对于整个序列的询问, 有一个非常简单的线段树做法:

- 对于右端点  $q$  做扫描线, 并利用单调队列快速维护对于不同的左端点  $p$ ,  $\max_{p \leq i \leq q} a[i]$  和  $\max_{p \leq j \leq q} b[j]$  的值。

假定  $A_{p,q} = \max_{p \leq i \leq q} a[i]$ ,  $B_{p,q} = \max_{p \leq j \leq q} b[j]$  的值。由于序列  $A[*, q]$ ,  $B[*, q]$  相较于  $A[*, q-1]$ ,  $B[*, q-1]$  都可以被看作是执行了若干次区间加得到的结果, 因此我们可以简单地维护一棵线段树, 使得其支持如下操作: 维护两个向量的点积。

- 对于单个向量进行区间加。

这是一个非常老套的经典线段树问题, 我们可以采用如下方式进行维护。当扫描到右端点  $q$  的时候, 对于区间  $[l, r]$ , 我们维护:

两个向量的区间加标记  $tA, tB$ ;

两个向量的区间和  $sA = \sum_{p=l}^r A_{p,q}$ ,  $sB = \sum_{p=l}^r B_{p,q}$ ;

区间包含的元素个数  $n = r - l + 1$ ;

两个向量在该区间内的点积  $d = \sum_{p=l}^r A_{p,q} B_{p,q}$ 。

通过对于上述标记以及区间信息的维护, 我们可以在右端点扫描到  $q$  的时候, 对于任意  $p$  询问  $\sum_{p=l}^r A_{p,q} B_{p,q}$  的值。因此可以通过边做扫描线, 边做区间求和的方式求解。

上述算法复杂度为  $O(Qn \log n)$ 。期望得分 32 分, 结合解法二期望得分 52 分。



### 【解法六】

正解做法:

基于解法五稍做修改. 如果我们能够在扫描到右端点  $q$  的时候, 在结点  $[l, r]$  内同时维护出  $d = \sum_{i=l}^r \sum_{j=i}^r A_i B_j$  的值, 则我们只需要在扫描到右端点的时候进行一次区间询问即可. 问题便转化为如何在已有信息上维护  $sd$ .

在维护两个向量的区间点积  $d$  的时候我们只运用到了区间元素个数  $n$ , 两个向量的区间点积  $d$  和两个向量的区间和  $sA, sB$ . 同时如果我们将区间加标记  $tA, tB$  视作常数, 则维护区间点积  $d$  的时候最多只运用到了上述元素的一次表达式. 同时由于  $sd$  可以被看成是对于不同的右端点的  $d$  求和, 因而我们可以尝试维护一个一次表达式作为  $sd$  的懒标记.

我们在每个结点  $[l, r]$  额外维护一个四元一次函数  $g(d, sA, sB, n) = kd * d + kA * sA + kB * sB + kn * n + kC$  作为标记, 其存储了如果我们将当前结点已经更新过  $sd$ , 但是没有下传到子结点的操作下传到子结点上, 在使用子结点更新过了的  $d, sA, sB, n$  的值作为输入的情况下, 子结点对应区间的  $sd$  改变量的表达式.

在每一次扫描右端点  $q$  的时候, 如果我们处理完所有区间加操作, 在结点  $[l, r]$  区间点积  $d$  已经存储了正确的值  $d = \sum_{i=l}^r A_i B_i$ , 且其正好是当前的  $sd$  和扫描到  $q-1$  时候的  $sd$  的差. 因此我们直接在根结点上打一个  $kd$  加一的标记, 并且直接将  $sd$  加上  $d$  即可. 不难发现该操作符合关于函数  $g$  的定义.

在下传标记的时候根据定义我们要首先下传区间加标记, 再下传  $g$  函数标记. 下传区间加标记的时候由于定义中我们利用的是已经更新过的  $d, sA, sB, n$  的值, 但是儿子结点并没有进行父结点区间加标记对应的更新, 因此对于儿子处较老的标记我们需要首先对其进行相应的变量代换. 利用父亲处的区间加标记更新儿子处  $g$  函数的标记后, 便可以直接和父亲的  $g$  函数标记进行逐项相加. 这样便完成了上述信息的维护.

上述所有的 6 个标记、5 个维护信息实际上都在同一个线段树框架下, 一次只需要使用线段树维护上述的所有 11 个标记即可. 时间复杂度  $O(n \log n + Q \log n)$ , 期望得分 100 分.

### 【总结】

本题的思维难度较大, 但是在理清楚思路后只需要按照描述维护上述标记即可. 同时本体的弱化版曾经出现在若干年前的湖南省选中, 因此部分选手会较容易地想到与当时题目解法更加类似的解法三和解法四的思路上去, 遗憾地错失了正确思路.

由于本题命制时间极早, 导致从命制完成到最终使用的这段时间内, 在网上出现过类似的题目. 这导致最终接触题目的选手比预期更多. 但是受限于前面几题耗时较长, 导致较少有同学能够有时间获取暴力分. 因而本题最终得分分布低于预期.

总体来说, 本题在 NOIP 的所有题目中思维难度略高, 但是考查知识点较为单一, 之后的命题应当减少类似题目.