2　arc068F Solitaire

2.1　题目链接

arc068.contest.atcoder.jp/tasks/arc068_d

2.2　题目大意

依次把$1 \dots n$放入一个双端队列的队首或队尾，然后进行n次取数，每一次从队首或队尾取一个元素。把每次取出的数按顺序排成一个序列。问能够构造出多少种不同的序列使得第k个元素为1？

2.3　数据范围

$1 \le k \le 2000$

2.4　性质分析

什么是满足条件的序列$a_{1 \dots n}$呢？

1. $a_k = 1$
2. 序列的前缀$a_{1 \dots k-1}$可以被划分成至多两个下降子序列
3. 对于$\forall i \in [k+1, n]$满足$a_i > \max(a_{i+1 \dots n})$或$a_i < \min(a_{i+1 \dots n})$
4. 存在一种方案，使得前缀$a_{1 \dots k-1}$划分成的两个不下降子序列中某一个子序列的最小值要大于$\max_{i=k+1}^{n} a_i$

Proof.

1. 第一条是题目给出的条件。

2. 对于第二条，因为一个数能被队列中取出必须满足：要么它左边的数全部被取出，要么它右边的数全部被取出。由于1号元素第一个被放入队列，其它数在1之后被按照从小到大的顺序放入队列两边，所以取出的时候一定是按照从大到小取出的。

3. 对于第三条，因为把数1取出来后队列剩下的部分是一个单调的数组，而每次只能取队首或队尾，所以每次只能取剩下数的最大值或最小值。

4. 对于第四条，我们假设1是从队列首部取出来的，那么我们可以用这张图大致描述队列的形态：

其中红色部分和黄色部分在1之前被取出，蓝色部分在1之后被取出。由于蓝色部分一定小于黄色部分，所以，把前缀$a_{1\ldots k-1}$划分成的两个不下降子序列中某一个子序列的最小值要大于$\max_{i=k+1}^{n} a_i$

从队尾取出来时的情况也同理。

□

2.5 DP

首先我们已经分析出了1之前序列能够划分成两个不下降子序列，那么我们可以进行DP，但是这题要求的是不同的子序列数目，所以我们要把一个合法前缀$a_{1\ldots k-1}$唯一对应两个不下降子序列。这里我们用的是一种贪心的方式，假设当前两个不下降子序列的结尾分别为$a, b(a < b)$，那么新加入一个数k后，如果$k < a$，那么我们把它放入第一个子序列的结尾，如果$a < k < b$，那么我们把它放入第二个子序列的结尾。并且，如果这个前缀合法，那么用这种贪心的方式最后一定能够构造出来。(对于$a < k < b$的情况，只能把数放入第二个子序列。对于$k < a$的情况，如果放入第一个子序列，那么接下来两个子序列的结尾分别是k, b，如果放入第二个子序列，那么两个子序列的结尾分别是a, k，而由于$a < b$，所以前者一定更优)。

然后我们用$f[i][j][low]$表示第一个子序列已经有i个元素，第二个子序列已经有j个元素，并且第一个子序列的末尾为low的方案数。由于我们每次贪心地把数优先分配给第一个子序列，所以第一个子序列的末尾一定<第二个子序列的末尾。所以此时选择的数都$\geq low$，那么有$n - low + 1 - i - j$个数$> low$且没有被选择。转移时，我们有两种方案：要么选择一个$> low$的数，转移到$f[i][j+1][low]$，要么选择一个$< low$的数，转移到$f[i+1][j][1\ldots low-1]$，用前缀和优化可以做到$O(n^3)$。

我们DP到序列的第k项就可以了，由于第k项为1，所以我们会贪心地把这个数分配给序列1。那么方案数就是$f[i][j][1](i+j=k)$。然后，把数字1从队列中取出之

后，剩下的数排列成什么形态题目已经没有限制，所以假设队列中还剩有x个数，那么每次从队尾或队首取出都是可以的，只要队列中元素个数≥2，从队首和队尾取出都会形成不同的序列，故方案数就为$2^{\max(x-1,0)}$。

2.6 优化

我们来优化这个DP的状态量，刚刚的DP需要记录一个序列1的最小值，我们不妨换一个顺序，改为依次把数字$n\dots 1$分配给两个序列，还是用刚刚那种贪心的思路保证每种方案被唯一计算。然后注意到，如果当前有一个数x 将分配给序列2，但是x比序列1的末尾元素小，那么此时我们不能马上放入取数序列，否则这将违背我们贪心的原则。而当序列1中出现了比x更小的数，那么第一个序列的最小值就会$< x$，此时我们就可以把x加入到取数序列中了。所以用$f[i][j][t \in \{1,2\}]$表示已经把$n\dots i$分配给了两个序列，还有j个数没有放入取数序列，且数字i被分配给了序列t的方案数。转移时，如果把数$i-1$分配给序列2，则这个数暂时还不能放入取数序列，转移到$f[i-1][j+1][2]$，如果把数$i-1$分配给序列1，则转移到$f[i-1][j][1]$。同时，如果当前数转移给了第一个序列，那么这j个未被放入的数都可以放入序列中了，所以我们可以对$f[i][j][1]$做关于j的后缀和就是总方案数，最后取$f[1][n-k]$就是把前缀$a_{1\dots k}$都安排好数的方案数，再乘上$2^{\max(x-1,0)}$即可。