

D. Doremy's Pegging Game

Problem – 1764D – Codeforces

Doremy has $n + 1$ pegs. There are n red pegs arranged as vertices of a regular n -sided polygon, numbered from 1 to n in anti-clockwise order. There is also a blue peg of **slightly smaller diameter** in the middle of the polygon. A rubber band is stretched around the red pegs.

Doremy is very bored today and has decided to play a game. Initially, she has an empty array a . While the rubber band does not touch the blue peg, she will:

1. choose i ($1 \leq i \leq n$) such that the red peg i has not been removed;
2. remove the red peg i ;
3. append i to the back of a .

Doremy wonders how many possible different arrays a can be produced by the following process. Since the answer can be big, you are only required to output it modulo p . p is guaranteed to be a prime number.

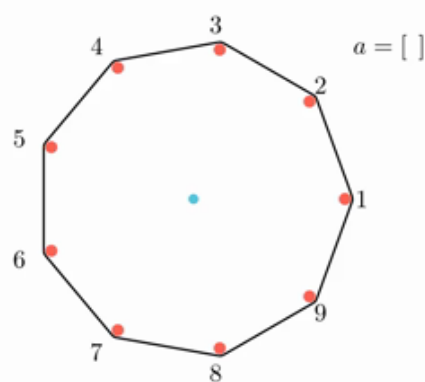
多雷米有 $n + 1$ 个图钉。有 n 个红色图钉，它们被排列成一个规则的 n 边多边形的顶点，按逆时针顺序从 1 到 n 依次编号。多边形中间还有一个直径**略小的蓝色图钉。一根橡皮筋缠绕在红色图钉上。

多雷米今天非常无聊，决定玩一个游戏。起初，她有一个空数组 a 。当橡皮筋没有碰到蓝色图钉时，她会碰到蓝色图钉：

1. 选择 i ($1 \leq i \leq n$)，这样红色图钉 i 就没有被移走；
2. 移除红钉 i ；
将 i 追加到 a 的后面。

多雷米想知道下面的过程可以产生多少个不同的数组 a 。因为答案可能很大，所以只需要输出它的模数 p 。保证 p 是质数。

有 $n = 9$ 和 $a = [7, 5, 2, 8, 3, 9, 4]$ 的游戏和另一个有 $n = 8$ 和 $a = [3, 4, 7, 1, 8, 5, 2]$ 的游戏



♪³ solve

参考博客：

CF1764D题解 – honglan0301's blog – 洛谷博客 (luogu.com.cn)

观察现象：

1. 首先第一个观察，一旦有一段长度 $\geq \frac{n}{2}$ 的点被删去，这种变化就会来到终点：
2. 于是问题转变成了一个选点的问题：

状态定义

$f_{i,j}$ 表示从删去连续的一段 1 到 $1 + i$ 个点，共删掉 j 个点的方案数数：

状态计算（就是每一个计数问题，而不是一个动态规划问题。）

1. 首先，明确最后一步删除必定 $2, 1 + \frac{n}{2}$ 以及 $i + 1 - \frac{n}{2}, i$ 的交中的。
2. i 大于等于 $\frac{n}{2}$ 才有贡献，这里的除法指的是向下取整。
3. 除了最后一个点，顺序是任意选择的。
4. 由于圆上处理的对称性质，只要求出一种情况最后通过 $\times n$ 就可以求出总答案了。

tips

1. 还漏了当 n 为偶数时的一种特殊情况：这种情况下，最后一条链的情况可以退化成一个点。

```

1  const int N = 5E3 + 10;
2
3  ll n;
4  const int C_maxn = 5010;
5  ll C[C_maxn][C_maxn];
6  ll fac[N];
7  void init(const int MOD)
8  {
9      C[0][0] = 1;
10     fac[0] = 1;
11     for (int i = 1; i < C_maxn; i++) {
12         fac[i] = fac[i - 1] * i % MOD;
13     }
14
15     for (int i = 1; i < C_maxn; i++) {
16         C[i][0] = 1;
17         C[i][i] = 1;
18         for (int j = 1; j < i; j++) {
19             C[i][j] = (C[i - 1][j] + C[i - 1][j - 1]) % MOD;
20         }
21     }
22 }
23 ll f[N][N], ans;
24 signed main() {
25     ios::sync_with_stdio(false);
26     cin.tie(0);
27     int p;
28     cin >> n >> p;
29     const int mod = p;
30     init(mod);
31     for (int i = n / 2 + 1; i < n; i++) {
32         for (int j = i - 1; j <= n - 2; j++) {
33             f[i][j] = C[n - i - 1][j - i + 1];
34             f[i][j] = f[i][j] * fac[j - 1] % mod;

```

```
35         f[i][j] = f[i][j] * (n / 2 + 1 - (i - n / 2)) % mod;
36         ans = (ans + f[i][j] * n % mod) % mod;
37     }
38 }
39 if (n % 2 == 0) {
40     ans = (ans + fac[n - 2] * n % mod) % mod;
41 }
42 cout << ans << "\n";
43 }
```

一个明智地追求快乐的人，除了培养生活赖以支撑的主要兴趣之外，总得设法培养其他许多闲情逸致。