

# Acwing 3195. 有趣的数

基于数学推导的快速解法，10ms AC

# 题目详情

## 3195. 有趣的数

📖 题目

📝 提交记录

💬 讨论

📖 题解

我们把一个数称为有趣的，当且仅当：

1. 它的数字只包含 0, 1, 2, 3，且这四个数字都出现过至少一次。
2. 所有的 0 都出现在所有的 1 之前，而所有的 2 都出现在所有的 3 之前。
3. 最高位数字不为 0。

因此，符合我们定义的最小的有趣的数是 2013。

除此以外，4 位的有趣的数还有两个：2031 和 2301。

请计算恰好有  $n$  位的有趣的数的个数。

由于答案可能非常大，只需要输出答案除以  $10^9 + 7$  的余数。

输入格式

输入只有一行，包括恰好一个正整数  $n$ 。

输出格式

输出只有一行，包括恰好  $n$  位的整数中有趣的数的个数除以  $10^9 + 7$  的余数。

数据范围

$4 \leq n \leq 1000$

输入样例：

4

输出样例：

3



# 解题思路

- 状态机
  - Acwing题解区基本是这个思路
  - 我觉得很好，如果不会状态机，一定要学一下
- 枚举 + 数学推导
  - 不过我的第一反应是枚举，而且结果非常不错，代码十分简洁（核心代码就三行），性能很高  $O(n)$
  - 本文稿主要讲解枚举思路，状态机思路可以访问：动态规划 + 代码注释 - AcWing - <https://www.acwing.com/solution/content/35638/>

## 3195. 有趣的数

题目

提交记录

讨论

题解

我们把一个数称为有趣的，当且仅当：

1. 它的数字只包含 0, 1, 2, 3，且这四个数字都出现过至少一次。
2. 所有的 0 都出现在所有的 1 之前，而所有的 2 都出现在所有的 3 之前。
3. 最高位数字不为 0。

因此，符合我们定义的最小的有趣的数是 2013。

除此以外，4 位的有趣的数还有两个：2031 和 2301。

请计算恰好有  $n$  位的有趣的数的个数。

由于答案可能非常大，只需要输出答案除以  $10^9 + 7$  的余数。



# 枚举解法

- 枚举的核心就是暴力确定任何一种可能出现的情形，并尽可能地降低时间复杂度。幸运的是，这道题，两个性质都满足。
- 枚举不能有遗漏，所以首先要考虑怎么枚举能够涵盖所有的可能，或者更准确地，先对枚举的情形进行分类，把各种情形放入一个个 bucket 里，然后一个一个 bucket 的汇总。那么，我们的 bucket 怎么选取呢？
- 可以考虑**每个字符第一次出现的位置**，见图右。这样一共就有三种可能，准确地说 3 的位置仅有三种可能：在 0 和 1 之前、在 0 和 1 之间、在 0 和 1 之后。
- 但是，如果我们看每个区间内值的可能，会发现，**除了每种字符第一次出现的单元格外，其他的单元格里，在 0 之前都只有一种可能，在 0 之后都会有两种可能。**
- 这是个非常重要的信息，也就是说，给定了 0、1、3 首次出现的位置后，该 bucket 对应的所有情形一共就是  $2^k$ ，其中  $k$  是位于 0 后面除去首次出现字符的剩余单元格数。
- 于是乎，我们可以合并 bucket 2 和 bucket 3，只考虑 3 在 0 前和 3 在 1 后的两种可能，并很快就能写出数学推导公式。
- 记 0 首次出现的位置为  $k$ （下标从 0 开始），则  $k \in [1, n - 2]$ ，因为至少有一个 1 在 0 之后。接下来，bucket 1 的数量有  $(k - 1) \times (n - 1 - k)$ （组合思想），bucket 2 和 bucket 3 的数量合计有  $(n - 1 - k) \times (n - 2 - k)$ （排列思想），所以对于任意  $k$ ，所有数字的组合就是
$$f(k) = (k - 1) \times (n - 1 - k) \times 2^{n-k-2} + (n - 1 - k) \times (n - 2 - k) \times 2^{n-k-3}$$
$$= (n - 1 - k) \times 2^{n-k-3} \times (n + k - 4), \text{ let } K = n - k - 2, K \in [0, n - 3]$$
$$= (K + 1) \times 2^{K-1} \times (2n - k - 6)$$
- 当  $K = 0$  时， $f(k) = n - 3$ ，这可以作为我们程序中 *ans* 的初始值，然后  $K$  从 1 开始递增，并使用一个  $m$  变量记录  $2^{K-1}$  项，递推以免去使用快速幂。

bucket 1

2	..2..	3	..3..	0	..3/0..	1	..3/1..
---	-------	---	-------	---	---------	---	---------

bucket 2

2	..2..	0	..2/0..	3	..0/3..	1	..3/1..
---	-------	---	---------	---	---------	---	---------

bucket 3

2	..2..	0	..2/0..	1	..2/1..	3	..1/3..
---	-------	---	---------	---	---------	---	---------



# 枚举的程序实现

```
1  #include "cstdio"
2  const int M = 1e9 + 7;
3
4  int main()
5  {
6      int n;
7      scanf("%d", & n);
8      int ans = n - 3;
9      long long m = 1;
10     for(int k=1; k <= n-3; k++)
11         ans += (k + 1) * (2 * n - 6 - k) * m % M,
12         ans %= M,
13         m <<= 1,
14         m %= M;
15     printf("%lld", ans);
16 }
17
```

作者:  南川

通过了 10/10个数据

运行空间: 216 KB

结果: Accepted

运行时间: 10 ms

提交时间: 1小时前

Thanks, feel free to feedback via  
[877210964@qq.com](mailto:877210964@qq.com) or 公众号: 南川笔记