








[Codeforces 1228E] Another Filling the Grid

 Siyuan (<https://blog.orzsiyuan.com/author/1/>)
  2019 年 10 月 01 日
  745 次浏览
  6 条评论
 4932 字数
  题解 (<https://blog.orzsiyuan.com/category/Problem/>)

 首页 (<https://blog.orzsiyuan.com/>) / 正文

分享到:  / ([http://sns.qzone.qq.com/cgi-bin/qzshare/cgi_qzshare_onekey?url=https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/&title=\[Codeforces 1228E\] Another Filling the Grid&site=https://blog.orzsiyuan.com/](http://sns.qzone.qq.com/cgi-bin/qzshare/cgi_qzshare_onekey?url=https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/&title=[Codeforces%201228E]%20Another%20Filling%20the%20Grid&site=https://blog.orzsiyuan.com/))

url=https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/&title=[Codeforces 1228E] Another Filling the Grid

 ([http://service.weibo.com/share/share.php?url=https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/&title=\[Codeforces 1228E\] Another Filling the Grid](http://service.weibo.com/share/share.php?url=https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/&title=[Codeforces%201228E]%20Another%20Filling%20the%20Grid))

题目链接: [Codeforces 1228E \(https://codeforces.com/contest/1228/problem/E\)](https://codeforces.com/contest/1228/problem/E)

你有一个 $n \times n$ 的网格和一个整数 k ，在每个格子中都填入一个整数，满足如下条件：

- 所有格子中的整数都介于 1 到 k 之间。
- 第 i 行的最小值为 1 ($1 \leq i \leq n$)。
- 第 j 列的最小值为 1 ($1 \leq j \leq n$)。

请求出填数的方案数，答案对 $10^9 + 7$ 取模。

数据范围: $1 \leq n \leq 250, 1 \leq k \leq 10^9$ 。

Solution

算法 1

设 $f(i, j)$ 表示已经填完了前 i 行，有 j 列已经包含 1 了。枚举上一行填了 k 个 1 则有：

- $f(0, 0) = 1$ 。
- $f(i, j) = f(i - 1, j) \cdot (k^j - (k - 1)^j) \cdot (k - 1)^{n-j}$ ，其中第二项表示这 j 个位置至少需要有一个 1，则可以用全集减去补集表示。
- $f(i, j) = f(i - 1, k) \cdot \binom{n-k}{j-k} \cdot (k - 1)^{n-j}$ 其中 $0 \leq k < j$ 。

时间复杂度: $\mathcal{O}(n^3)$ 。

算法 2

考虑容斥。我们枚举至少 i 行和 j 列没有 1。那么答案为：

$$\sum_{i=0}^n \sum_{j=0}^n (-1)^{i+j} \binom{n}{i} \cdot \binom{n}{j} \cdot (k-1)^{n(i+j)-ij} \cdot k^{(n-i)(n-j)} \quad (1)$$

时间复杂度: $\mathcal{O}(n^2) \sim \mathcal{O}(n^2 \log n)$ 。

算法 3

我们对 (1) 进行化简，具体方法为：对第二重 j 求和根据二项式定理展开，最终得到的式为：

$$\sum_{i=0}^n (-1)^i \cdot \binom{n}{i} \cdot (k^{n-i} \cdot (k-1)^i - (k-1)^n)^n$$

时间复杂度: $\mathcal{O}(n \log n)$ 。

Code

算法 1

```
1  #include <cstdio>
2
3  const int N = 250;
4  const int MOD = 1e9 + 7;
5
6  int n, k, p[N + 5], q[N + 5], c[N + 5][N + 5], f[N + 5][N + 5];
7
8  void add(int &x, int y) {
9      (x += y) >= MOD && (x -= MOD);
10 }
11 void sub(int &x, int y) {
12     (x -= y) < 0 && (x += MOD);
13 }
14 int add(int x) {
15     return x >= MOD ? x - MOD : x;
16 }
17 int sub(int x) {
18     return x < 0 ? x + MOD : x;
19 }
```

算法 2

```
1  #include <cstdio>
2
3  const int N = 250;
4  const int MOD = 1e9 + 7;
5
6  int n, k, fac[N + 5], ifac[N + 5], pw1[N * N + 5], pw2[N * N + 5];
7
8  void add(int &x, int y) {
9      (x += y) >= MOD && (x -= MOD);
10 }
11 void sub(int &x, int y) {
12     (x -= y) < 0 && (x += MOD);
13 }
14 int pow(int x, int k) {
15     int ans = 1;
16     for (; k > 0; k >>= 1, x = 1LL * x * x % MOD) {
17         if ((k & 1) == 1) ans = 1LL * ans * x % MOD;
18     }
19     return ans;
```

算法 3

```
1  #include <cstdio>
2
3  const int N = 250;
4  const int MOD = 1e9 + 7;
5
6  int n, k, fac[N + 5], ifac[N + 5], p[N + 5], q[N + 5];
7
8  void add(int &x, int y) {
9      (x += y) >= MOD && (x -= MOD);
10 }
11 void sub(int &x, int y) {
12     (x -= y) < 0 && (x += MOD);
13 }
14 int mod(int x) {
15     return x < 0 ? x + MOD : x >= MOD ? x - MOD : x;
16 }
17 int pow(int x, int k) {
18     int ans = 1;
19     for (; k > 0; k >>= 1, x = 1LL * x * x % MOD) {
```

本作品采用 知识共享署名-相同方式共享 4.0 国际许可协议 (<http://creativecommons.org/licenses/by-sa/4.0/>) 进行许可。

下一篇 (<https://blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/>)

上一篇 (<https://blog.orzsiyuan.com/archives/%E5%A5%BD%E5%BA%B7%E7%9A%84/>)

6 条评论



lihaoxiang2006

November 8th, 2019 at 21:57

qwq

回复 (<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/?replyTo=348#respond-post-1120>)



Schwarzkopf_Henkal (<https://www.cnblogs.com/schwarzkopf-henkal/>)

November 6th, 2019 at 09:08

代码部分的样式是用什么实现的?

回复 (<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/?replyTo=340#respond-post-1120>)

**Siyuan** (<http://orzsiyuan.com>) 博主

November 6th, 2019 at 12:38

@Schwarzkopf_Henkal 插件 CodePrettify回复 (<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/?replyTo=343#respond-post-1>)**dd**

October 7th, 2019 at 08:28

技术的路过。

回复 (<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/?replyTo=323#respond-post-1120>)**dd**

October 7th, 2019 at 08:28

cai

回复 (<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/?replyTo=322#respond-post-1120>)**repostone** (<https://repostone.home.blog/>)

October 3rd, 2019 at 16:06


非技术的路过。

回复 (<https://blog.orzsiyuan.com/archives/Codeforces-1228E-Another-Filling-the-Grid/?replyTo=317#respond-post-1120>)

发表评论

评论 *

说点什么吧.....

 表情私密评论 ☐

名称 *



姓名或昵称

邮箱 *

邮箱 (必填,将保密)

地址

网站或博客

发表评论



热门文章

- (<https://blog.orzsiyuan.com/archives/ZJOI-2019/>) (<http://chilogs.zjoi.org.cn/2019/>)

ZJOI-2019 游记 (http://chilogs.zjoi.org.cn/2019/)

2019/)

6051
- (<https://blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/>)

CSP-2019 算法模板复习 (https://hehezhou.blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/)

AK-

2892
- (<https://blog.orzsiyuan.com/archives/Polynomial-Template/>)

多项式模板 (https://blog.orzsiyuan.com/archives/Polynomial-Template/)

2019/)

1080
- (<https://blog.orzsiyuan.com/archives/SDOI-2017-Number-Table/>)

SDOI-2017 数字表格 (https://SDOI.blog.orzsiyuan.com/archives/SDOI-2017-Number-Table/)

2017-

1028
- (<https://blog.orzsiyuan.com/archives/TJOI-2019-Sing-Dance-Rap-and-Basketball/>)

TJOI-2019 唱歌、跳舞和篮球 (https://blog.orzsiyuan.com/archives/TJOI-2019-Sing-Dance-Rap-and-Basketball/)

2019-

843

博客信息

文章数目	187
评论数目	243
运行天数	1年25天
最后活动	4 个月前