

「Codeforces 1204E」 Natasha, Sasha and the Prefix Sums

● Siyuan (<https://blog.orzsiyuan.com/author/1/>) ⊖ 2019 年 09 月 08 日 眼 622 次浏览 ↗ 1 条评论
 ✎ 3344 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-1204E-Natasha-Sasha-and-the-Prefix-Sums/&title=%E2%80%9CCodeforces-1204E%E2%80%9D-Natasha,-Sasha-and-the-Prefix-Sums>

<http://service.weibo.com/share/share.php?url=https://blog.orzsiyuan.com/archives/Codeforces-1204E-Natasha-Sasha-and-the-Prefix-Sums&title=%E2%80%9CCodeforces-1204E%E2%80%9D-Natasha,-Sasha-and-the-Prefix-Sums>

url=<https://blog.orzsiyuan.com/archives/Codeforces-1204E-Natasha-Sasha-and-the-Prefix-Sums/&title=%E2%80%9CCodeforces-1204E%E2%80%9D-Natasha,-Sasha-and-the-Prefix-Sums>

题目链接: [Codeforces 1204 \(https://codeforces.com/contest/1204/problem/E\)](https://codeforces.com/contest/1204/problem/E)

Natasha 最喜欢的数字是 n 和 1, Sasha 最喜欢的数字是 m 和 -1 。某一天他们写下了长度为 $n + m$ 且包含恰好 n 个 1 和 m 个 -1 的所有可能的序列。对于每一个序列计算出它的最大前缀和（允许为空）；形式化地，我们定义 $f(a)$ 表示序列 $a_1, \dots, a_l (l \leq 0)$ 的最大前缀和，那么有：

$$f(a) = \max \left(0, \max_{i=1}^l \sum_{j=1}^i a_j \right)$$

现在他们想要对于所有满足条件的序列，求出 $f(a)$ 的总和。答案对 998244853 取模。

数据范围： $0 \leq n, m \leq 2000$ 。

Solution

弱化版

我们设 $f(i, j)$ 表示含有 i 个 1 和 j 个 -1 的所有序列的 $f(a)$ 之和。为了体现「前缀和」的性质，我们每次加入一个数字时不是加在序列的最后面，而是加在最前面。因为我们发现这样可以使新的数字对所有前缀和造成影响；否则无法计算最大前缀和是否被修改。

为了方便描述，我们定义 $g(i, j)$ 表示含有 i 个 1 和 j 个 -1 的序列的个数。显然有：

$$g(i, j) = \binom{i+j}{i}$$

对于 f 有转移：

$$\begin{aligned} f(i, j) + 1 \cdot g(i, j) &\rightarrow f(i+1, j) \\ f(i, j) + (-1) \cdot (g(i, j) - h(i, j)) &\rightarrow f(i, j+1) \end{aligned}$$



第二个转移中的 $h(i, j)$ 是什么鬼？如果原序列的 $f(a) = 0$ ，那么在原序列最前面加上一个 -1 后不会影响其 f 值。这样一来我们就需要记录 $h(i, j)$ 表示含有 i 个 1 和 j 个 -1 的序列中最大前缀和为 0 的方案数，其转移为（考虑在序列末尾加入新数字）：

$$h(i, j) = \begin{cases} 1 & i = 0 \\ 0 & i > j \\ h(i-1, j) + h(i, j-1) & \text{otherwise.} \end{cases}$$

题外话：我们可以发现 h 的定义和「卡特兰数」类似，则有 $h(i, j) = \binom{i+j}{j} - \binom{i+j}{j+1}$ 。

时间复杂度： $O(nm)$ 。

加强版

如果我们把数据范围加强到 $0 \leq n, m \leq 10^7$ 呢？~~夫毒瘤！~~

我们对于每种答案统计方案数。设最大前缀和为 i 的方案数有 $f(i)$ 种，我们不妨对其做后缀和得到：最大前缀和不小于 i 的方案数有 $g(i)$ 种。

我们参考卡特兰数的几何证明，将 1 看做平面向量 $(1, 1)$ 、将 -1 看做平面向量 $(1, -1)$ ，那么问题等价于：**从点 $(0, 0)$ 出发到达点 $(n+m, n-m)$ 的所有路径中，经过直线 $y = i$ 的方案数。**

对于 i 分两种情况讨论：

1. 当 $0 \leq i \leq n-m$ 时，起点和终点在直线的两侧，则 $g(i) = \binom{n+m}{n}$ 。
2. 当 $n-m < i \leq n$ 时，我们找到路径上第一个经过 $y = i$ 的点，将起点到这个点的路径通过 $y = i$ 对称，起点通过变换变成了 $(0, 2i)$ 。设路径上有 x 个 $(1, 1)$ 和 y 个 $(1, -1)$ ，列出方程：

$$\begin{cases} x+y = n+m \\ x-y = n-m-2i \end{cases}$$

得到 $x = n - i, y = m + i$, 方案数 $g(i) = \binom{x+y}{x} = \binom{n+m}{n-i}$ 。

最终的答案为

$$\sum_{i=0}^n i \cdot (g(i) - g(i+1))$$



时间复杂度: $O(n + m + \log p)$ 。

Code

此处只给出**加强版**的代码。

```

1 #include <cstdio>
2
3 const int N = 2e7;
4 const int MOD = 998244853;
5
6 int n, m, fac[N + 5], ifac[N + 5], f[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 }
```

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 \(https://blog.orzsiyuan.com/archives/Codeforces-662C-Binary-Table/\)](https://blog.orzsiyuan.com/archives/Codeforces-662C-Binary-Table/)[上一篇 \(https://blog.orzsiyuan.com/archives/Codeforces-1217D-Coloring-Edges/\)](https://blog.orzsiyuan.com/archives/Codeforces-1217D-Coloring-Edges/)

1 条评论

**xddd (<https://cdn.jsdelivr.net>)**

September 8th, 2019 at 19:54



Orz Siyuan

[回复 \(https://blog.orzsiyuan.com/archives/Codeforces-1204E-Natasha-Sasha-and-the-Prefix-Sums/?replyTo=304#respond-post\)](https://blog.orzsiyuan.com/archives/Codeforces-1204E-Natasha-Sasha-and-the-Prefix-Sums/?replyTo=304#respond-post)

发表评论 i

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

[\(https://blog.orzsiyuan.com/archives/ZJOI-2019/\)](https://blog.orzsiyuan.com/archives/ZJOI-2019/)

2019/ | 眼 6051

[\(https://blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/\)](https://blog.orzsiyuan.com/archives/hehezhou-AK-CSP-2019/)

AK- | 眼 2892

CSP-
2019/) (<https://blog.orzsiyuan.com/archives/Polynomial-Template/>)
Template) 1080

(<https://blog.orzsiyuan.com/archives/SDOI-2017-Number-Table/>)
2017- 眼 1026

Number-
Table/) (<https://blog.orzsiyuan.com/archives/TJOI-2019-Sing/>
2019- Dance-Rap-and-Basketball/) 843
Sing-
Dance-
Rap-
and-
Basketball/)

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 |

© 2020 Copyright 浙ICP备19008446号-1 (<http://www.beian.miit.gov.cn>)