

E3-G 题题解

王家豪

November 8, 2022

G 题题意

在一个 n 行 k 列的方格表中，你需要从第 1 行走到第 n 行，求不同的走法数。需满足：相邻的两行不能经过同一列，同时给出 m 个数 a_1, a_2, \dots, a_m ，这些行的第 k 列不能进入。

Observation

1	1	
1	1	2
3	3	
3	3	6

容易发现前 $k-1$ 列是“地位相等”的，计数方式相同，最后一列单独考虑即可。

核心思路

记 $dp1[i]$ 为第 i 行的前 $k-1$ 列的方法数之和, $dp2[i]$ 为第 i 行第 k 列的方法数, 则有

$$dp1[i] = (k-2) \cdot dp1[i-1] + (k-1) \cdot dp2[i-1]$$

$$dp2[i] = \begin{cases} dp1[i-1], & i \text{ not in a} \\ 0, & \text{otherwise} \end{cases}$$

核心代码

```
dp1[1] = k - 1;
dp2[1] = vis[1] ? 0 : 1;
for (int i = 2; i <= n; i++) {
    dp1[i] = (dp1[i - 1] * (k - 2) +
              dp2[i - 1] * (k - 1)) % mod;
    dp2[i] = (vis[i] ? 0 : dp1[i - 1]) % mod;
}
```

时间复杂度: $O(n)$

完整代码

```
#include <stdio.h>

#define N 1000005
#define mod 998244353

long long dp1[N], dp2[N];
int vis[N];

int main() {
    int n, m, k;
    scanf("%d%d%d", &n, &m, &k);
    for (int i = 1; i <= m; i++) {
        int a;
        scanf("%d", &a);
        vis[a] = 1;
    }
    dp1[1] = k - 1;
    dp2[1] = vis[1] ? 0 : 1;
    for (int i = 2; i <= n; i++) {
        dp1[i] = (dp1[i - 1] * (k - 2) + dp2[i - 1] * (k - 1)) % mod;
        dp2[i] = (vis[i] ? 0 : dp1[i - 1]) % mod;
    }
    printf("%lld\n", (dp1[n] + dp2[n]) % mod);
    return 0;
}
```