> C-杀戮尖塔
>
> 23371041-李一鸣

题目描述

- 游戏一共 n 层，每层 k 个关卡，相邻两层之间不能进入同一关
- 给定 m 和 m 个层，在给定的层中不能进入第 k 关
- 求打通 n 层的方案数

解题思路

- 考虑用动态规划求解
- 设 $dp_{i,j}$ 为 通关第 i 层，通关关卡为 j 的方案数
- 若层数 i 为给定的不能访问 k 关卡的层，则设 $vis(i) = 1$
- 则考虑以下转移方程:
  - 若 $j \neq k$，则状态可由其他 $k-1$ 个关卡转移过来
  - 若 $j = k$，则状态取决于 $vis(i)$
    - 若 $vis(i) = 0$，则状态可由其他 $k-1$ 个关卡转移过来
    - 否则为 0
  - 写出转移方程如下:
    - $$dp_{i,j} = \begin{cases} \sum\limits_{1\le l\le k, l\neq j} dp_{i-1,l}, & j\neq k \\ \sum\limits_{1\le l<k} dp_{i-1,l}, & j=k, vis(i)=0 \\ 0, & j=k, vis(i)=1 \end{cases}$$
- 由于 k 值过大，二维数组存放不下，因此考虑优化方程
- 不难发现，$1 \sim k-1$ 关是等价的，仅与第 k 关有所区分
- 因此，考虑优化状态表示：$dp_{i,j}(1 \le i \le n, j = 0,1)$ 表示通关第 i 层，通关关卡为 $1 \sim k-1$ 关或 第 k 关的方案数
- 得出转移方程如下:
  - 若 $j \neq k$，则状态可由 $1 \sim k-1$ 关和第 k 关转移而来
  - 否则，依旧考虑 $vis(i)$
    - 若 $vis(i) = 0$，则状态可由 $1 \sim k-1$ 关转移而来
    - 否则为0
  - $$dp_{i,j} = \begin{cases} dp_{i-1,0} \times (k-2) + dp_{i-1,1} \times (k-1), & j=0 \\ dp_{i-1,0}, & j=1, vis(i)=0 \\ 0, & j=1, vis(i)=1 \end{cases}$$
- 复杂度分析：由于 $1 \le n \le 10^6$，时间复杂度为 $\Theta(n)$，因此可以通过本题。

代码实现

```
int dp[1000010][2];
int n, m, k;
int barrier[1000010];
const int mod = 998244353;

signed main() {
    n = read(), m = read(), k = read();
    for(int i = 1; i <= m; i++) {
        int x = read();
        barrier[x] = 1;
    }
    // dp i_0 for case 1 ~ k - 1
    // dp i_1 for case k
    dp[1][0] = k - 1, dp[1][1] = barrier[1] ? 0 : 1;
    for(int i = 2; i <= n; ++i) {
        dp[i][0] = (dp[i - 1][0] * (k - 2)) % mod;
        dp[i][0] = (dp[i][0] + dp[i - 1][1] * (k - 1)) % mod;
        if(barrier[i])
            dp[i][1] = 0;
        else
            dp[i][1] = (dp[i - 1][0]) % mod;
    }
    int ans = (dp[n][0] + dp[n][1]) % mod;
    write(ans);
    return 0;
}
```