

# E2-D problem

## 题目描述

蛇皇帝扔给莫卡一个长度为  $n$  的序列  $a_1, a_2, \dots, a_n$ ，初始有  $a_i = i$ ，你需要维护一个数据结构，支持  $m$  次下列三种操作之一：

1. 给定正整数  $l, r$ ，对任意  $l \leq i \leq r$ ，将  $a_i$  变为  $a_i^2$ 。
2. 给定正整数  $l, r, k$ ，对任意  $l \leq i \leq r$ ，将  $a_i$  变为  $k \cdot a_i$ 。
3. 给定正整数  $l, r$ ，输出  $(\prod_{i=l}^r a_i) \bmod (r-l+1)!$ 。

## 题解思路

- 此题只要对于操作3全部输出0即可AC，比较偷鸡的解题方法。
- 具体原因：对于操作3，从  $a_l$  到  $a_r$  一共  $r-l+1$  个数，最后要模  $(r-l+1)!$ ，求余数。而  $a_l$  到  $a_r$  的初始取值便是整数  $l$  到  $r$ ，他们连乘起来一定能被  $(r-l+1)!$  整除。而操作1,2对于连乘的结果只是添加了若干个质因数，因此依旧能够保证连乘能被  $(r-l+1)!$  整除。故操作3输出结果全为0。

## 代码

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

#define MAX 1000005

long long box[MAX] = {0};

int main() {
    int n, m;
    scanf("%d %d", &n, &m);
    // for (int i = 1; i <= n; i++) {
    //     box[i] = i;
    // }
    while (m--) {
        int op;
        scanf("%d", &op);
        if (op == 1) {
            int l, r;
            scanf("%d %d", &l, &r);
            // for (int i = l; i <= r; i++) {
            //     box[i] *= box[i];
            // }
        }
        else if (op == 2) {
            int l, r, k;
            scanf("%d %d %d", &l, &r, &k);
            // for (int i = l; i <= r; i++) {
            //     box[i] *= k;
            // }
```

```
    }  
    else if (op == 3) {  
        long long l, r;  
        scanf("%lld %lld", &l, &r);  
        printf("0\n");  
    }  
}  
return 0;  
}
```