

C1-C problem

题目描述

小水獭正在学习模运算。

现在小水獭有 n 个非负整数 a_1, a_2, \dots, a_n 它定义 $f(i)$ 为除了 a_i 以外的 $n-1$ 个非负整数的乘积对 10^9+6 取模后的结果，形式化地说：

$$f(i) = \left(\prod_{j=1, j \neq i}^n a_j \right) \bmod (10^9 + 6)$$

小水獭希望你可以帮助它写一个程序计算 $f(1), f(2), \dots, f(n)$ 。

输入格式及数据规模

第一行一个正整数 t ($1 \leq t \leq 5$)，表示数据组数。

对于每组数据，第一行一个正整数 n ($2 \leq n \leq 10^5$)，含义同题目描述。

第二行 n 个非负整数 a_1, a_2, \dots, a_n ($0 \leq a_i < 10^9+6$)，含义同题目描述。

题解思路

- 为了避免计算过程中数值过大越界，利用提示： $ab \bmod c = (a \bmod c)(b \bmod c) \bmod c$ ，在每次计算两数乘法时先将两数分别取模相乘后再取模。
- 对于每个 $f(i)$ ，若按照公式直接计算模值，由于 n 的规模为 10^5 ，算法时间复杂度为 $O(n^2)$ ，实测会超时。
- 对算法进行优化：利用提示的将 $f(i)$ 分为前后两部分进行计算，同B题类似计算每项过程中可以利用前一项的结果，将算法的时间复杂度降低到了 $O(n)$
- 由于 a_i 大小的上限为 10^9+6 ，两数相乘时可能会超过 int 的最大范围，因此在计算余数时使用 long long 类型变量进行存储，避免越界。

代码

```
#include <iostream>
#include <math.h>
#include <vector>

using namespace std;

const int MOD = (int)(pow(10, 9) + 6);

int main() {
    int n = 0;
    cin >> n;
    vector<vector<long long>> v_2;
```

```

for (int i = 0; i < n; i++) {
    long long n_tmp;
    cin >> n_tmp;
    vector<long long> v_tmp;
    for (int j = 0; j < n_tmp; j++) {
        long long num_tmp;
        cin >> num_tmp;
        v_tmp.push_back(num_tmp);
    }
    v_2.push_back(v_tmp);
}

for (int i = 0; i < n; i++) {
    vector<long long> front;
    vector<long long> back;
    int len = v_2[i].size();
    front.push_back(1);
    back.push_back(1);
    // front.push_back(v_2[i][0]);
    // back.push_back(v_2[i][len-1]);
    for (int j = 1; j < v_2[i].size(); j++) {
        front.push_back((front[j-1] * v_2[i][j-1]) % MOD);
        back.push_back((back[j-1] * v_2[i][len-j]) % MOD);
    }
    for (int j = 0; j < v_2[i].size(); j++) {
        long long fmod = 1;
        fmod = (front[j] * back[len-1-j]) % MOD;
        if (j != v_2[i].size() - 1) {
            cout << fmod << ' ';
        }
        else {
            cout << fmod << endl;
        }
    }
}

return 0;
}

```