

C2-C题题解

题目描述

对于一个数组 $A = [a_1, a_2, \dots, a_n]$, 定义

$$f(A) = \sum_{i=1}^n \text{order}(a_i) \cdot a_i$$

其中

$$\text{order}(a_i) = 1 + \sum_{\substack{a_j < a_i \\ 1 \leq j \leq n}} 1$$

现给出一个随机数生成器如下:

```
#include <stdio.h>
#define N 5000005

int nextRand() {
    static unsigned int rnd_num = 0x80000001;
    static int mod = 1e5 + 3;

    rnd_num ^= rnd_num >> 10;
    rnd_num ^= rnd_num << 9;
    rnd_num ^= rnd_num >> 25;
    return rnd_num % mod;
}

int a[N];

int main() {
    int tt;
    scanf("%d", &tt);
    while (tt--) {
        int n;
        scanf("%d", &n);
        for (int i = 1; i <= n; i++) {
            a[i] = nextRand();
        }
    }
    return 0;
}
```

对于每组测试数据，你需要使用上述随机数生成器生成 n 个数 a_1, a_2, \dots, a_n ，记 $A = [a_1, a_2, \dots, a_n]$ ，你需要求出 $f(A)$ 的值。

输入

第一行一个正整数 T ($1 \leq T \leq 10$) 表示数据组数。

接下来 T 组数据：

一行一个正整数 n ($1 \leq n \leq 5 \cdot 10^6$) 表示数组 A 的长度。

输出

对于每组数据，输出一行一个正整数表示 $f(A)$ 的值。

解题

求解本题的关键就是求得小于某个数的数的个数

- 注意产生随机数的函数内的 `return` 语句，结果 `% mod`，说明产生的随机数的范围是 $0 \sim \text{mod}-1$

```
int nextRand() {
    static unsigned int rnd_num = 0x80000001;
    static int mod = 1e5 + 3;

    rnd_num ^= rnd_num >> 10;
    rnd_num ^= rnd_num << 9;
    rnd_num ^= rnd_num >> 25;
    return rnd_num % mod;
}
```

- 对于数组中的每个数，需要得到该数组中小于它的数的个数
 - 如果对数组进行**排序**，使其**递增**，那么该数组中小于它的数的个数就可以与其**下标**联系起来
 - 但是如果使用快排的话，时间复杂度为 $O(n \log n)$ ，那么本题的时间复杂度就是：
 - 数据组数 $10 * n \log n$ ，代入 $n = 5e6$ 时的情况，估算出结果大于 10^9 ，无法通过本题

假设时间限制为1秒

10^6	1000000	游刃有余
10^7	10000000	勉勉强强
10^8	100000000	很悬，仅限循环体非常简单的情况

- 回看本题的题目“**排序？数数！**”，指明了本题的思路应该是“**数数**”而不是“**排序**”
- 如何“数数”？
 - 定义数组 `int cnt[]` (`cnt` 是 `count` 的缩写)，`cnt[i]` 表示数组中**数字 i 出现的次数**
 - 所以想要得到小于 x 的数的个数，就只需要求 `cnt[0]+cnt[1]+.....+cnt[x-1]` 即可

- 但是对于数组中的每个数，如果我们都像这样求累加和来求小于它的数的个数的话，也是会超时的，此时我们需要用**前缀和**算法来简化这一步骤

前缀和

回想高中数学的数列知识，我们定义数列 a_n ，其前 n 项和为 S_n

那么 $a[0] + a[1] + \dots + a[x-1]$ 就是 $S[x-1]$

$a[L] + a[L+1] + \dots + a[R-1] + a[R]$ 就是 $S[R] - S[L-1]$ ($L \geq 1$)

即我们可以**直接访问**数组中**连续一段元素**的和

如果我们预先处理出来 $S[0]$ 一直到 $S[\text{mod}-1]$ ，那么**直接访问数组S**就可以得到某个数前面有多少个数

代码(C++)

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

using namespace std;
const int N = 5e6 + 5, MOD = 1e5 + 3;

int nextRand() {
    static unsigned int rnd_num = 0x80000001;
    static int mod = 1e5 + 3;

    rnd_num ^= rnd_num >> 10;
    rnd_num ^= rnd_num << 9;
    rnd_num ^= rnd_num >> 25;
    return rnd_num % mod;
}

int a[N];
// 答案会超出int范围 (int的最大正数大概是 $2 \times 10^9$ )，所以需要开long long
long long ans;

int cnt[MOD], S[MOD];

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        // 多组数据输入，需要注意全局变量的初始化以及清空
        // 把cnt和S数组都初始化为0
        memset(cnt, 0, sizeof cnt);
        memset(S, 0, sizeof S);
        ans = 0;

        int n;
        scanf("%d", &n);
```

```
for (int i = 1; i <= n; i++) {
    a[i] = nextRand();
    // 产生了随机数a[i], 则值为a[i]的数的数量+1
    cnt[a[i]] += 1;
}
// 预处理出s数组
for (int i = 1; i < MOD; i++) {
    S[i] = S[i - 1] + cnt[i];
}
for (int i = 1; i <= n; i++) {
    // 比a[i]小的数的数量就是S[a[i] - 1]
    ans += (1 + S[a[i] - 1]) * a[i];
}
cout << ans << endl;

}
return 0;
}
```