3月11日(Day 3)题解

目 录

T 1

计算题

题目大意

这是一道简单的计算题,给定 a, n 求:

$$(a-1)\times\sum_{i=0}^{n} a^i$$

答案可能很大,输出时对 10^9+7 取模

数据范围

对于 10% 的数据, $n \le 100$

对于 20% 的数据, $n \le 10^9$

对于 100% 的数据, $2 \le n \le 10^{100000}, 2 \le a \le 50$

解题思路

- 不难发现 $\sum_{i=0}^{n} a^i$ 是一个等比数列求和，可以把它化简为 $\frac{a^{n+1}-1}{a-1}$ 。 代入原式可以得到答案其实就是 $a^{n+1}-1$。
- 据~~费脑~~费马小定理可以得到 $a^{n+1} \bmod 10^9+7 = a^{(n+1)\bmod(10^9+6)} = a^{n \bmod(10^9+6)+1}$，
把n改写成 $\overline{a_1a_2......a_n}$ ， 可以得到 $n \bmod(10^9+6) =$
$(a_1*10^{n-1}+a_2*10^{n-2}+......+a_n*10^0)\bmod(10^9+6) =$
$a_1*10^{n-1}\bmod(10^9+6)+a_2*10^{n-2}\bmod(10^9+6)+......+a_n*10^0\bmod(10^9+6)$
- 利用快速幂计算即可。

代码片段

```
scanf("%lld%s", &a, s);
for (int i = 0; i < strlen(s); i++) {
    ans += ((s[i] - '0') % (MOD - 1)) *
        q_pow(10, strlen(s) - i - 1, MOD - 1);
    ans %= (MOD - 1);
}
ans++;
ans %= MOD;
printf("%lld", q_pow(a, ans, MOD) - 1);
```

T 2

完美稳定分割

题目大意

我们称一个分割整数数组的方案是完美的,当它满足:

数组被分成三个 **非空** 连续子数组,从左至右分别命名为 left,mid,right

$$\text{left} \le \text{mid} \le \text{right}$$

> left *中元素和小于等于* mid *中元素和,* mid *中元素和小于等于* right *中元素和*

给你一个 非负 整数数组 A ,请你输出完美的分割 A 方案数目

由于答案可能会很大,请你将结果对 1000000007 取模输出

数据规模

对于 30% 的数据 $1 \le n \le 100$

对于 50% 的数据 $1 \le n \le 5000$

对于 100% 的数据 $1 \le n \le 10^5$

对于全部的数据 $0 \le A_i \le 10^5$

解题思路

- 先求出数组的前缀和。接着枚举left的长度，再用2个二分分别求出left与mid的边界，mid与right的边界。注意处理无解的情况。
  - 找出边界后，利用加法原理计算即可（别忘记 mod！！！！）

```
for (i = 1; i < n - 1; i++) {
    int l = ef_find(i + 1, n - 1, q[i]);
    if (l == -1) continue;
    int plu = ef_ans(l, n - 1, q[i]);
    if (plu == -1) continue;
    ans += plu - l + 1;
    ans %= 1000000007;
}
```

q 是前缀和数组

T 3

差异素对

题目大意

diff-prime 问题是给定 N ,你需要找存在多少对 (i, j) 使得 $\frac{i}{\gcd(i, j)}$ 和 $\frac{j}{\gcd(i, j)}$ 都是素数,其中 $(1 \le i, j \le N)$.

请帮助艾迪解决这个问题

注意,如果 $i_1 \neq i_2$ 或 $j_1 \neq j_2$,则对 (i_1, j_1) 和对 (i_2, j_2) 被认为是不同的

数据规模

对于 10% 的数据 $1 \le N \le 1000$

对于 30% 的数据 $1 \le N \le 10000$

对于 60% 的数据 $1 \le N \le 10^6$

对于全部的数据 $1 \le N \le 10^7$

暴 力

不难想到一种暴力，O(N^2)枚举i, j，计算gcd(i, j)，判断是否符合条件，计数。

期望得分10分

优化一点点

先筛出N以内的质数，枚举i，j（i<=j）计数后将结果乘2。

期望得分30分

正 解

欧拉筛筛出N以内的质数，顺便用sum数组算出sum[i]=i以内的质数个数。从1~n枚举一遍gcd(i, j)，计数器计算sum[n/gcd(i, j)]中取2个数排列的方案数，即

sum[n/gcd(i, j)] * sum[n/gcd(i, j)]-1

```
oulashai(n);
for (int i = 1; i <= n; i++) {
    long long tem = num[n / i];
    ans += tem * (tem - 1);
}
```

T 4

爬山

题目大意

- 有一条长度为LEN的线段，一点p从线段左端点以v个单位每年向右端点移动。在线段上有n个技能点，分别在距左端点a[1], a[2]……, a[n]的位置。如果激活了技能点，那p点**第一次**到达这里时会**立即**回到起点。
- 给定q个时间t，问至少激活几个点才能使p点到达右端点的时间$>t$。若激活所有的技能点也无法做到这一点，输出-1。

数据规模

对于测试点 $1 \sim 2$,满足 $n = 1$

对于测试点 $3 \sim 5$,满足 $n = 2$

对于测试点 $6 \sim 10$,满足 $n, q \le 1000$

对于所有测试点,满足 $1 \le n, q \le 2 \times 10^5, 1 \le v \le L \le 10^9, 1 \le a_i < L, 1 \le t_i \le 10^9$

数据保证 a_i 两两不同

解题思路

- 首先将所有技能点降序排序（一定是先选离起点远的），然后求出每个技能点能使p点慢几秒，很明显这是个前缀和，用数组q[i]存第i个点能使p点慢几秒。
- 二分答案找出最少ans个点能使时间大于t

```
int n, v, T, i, tem;
scanf("%d%d%d", &n, &T, &v);
L = T * 1.0 / v;
for (i = 1; i <= n; i++) {
    scanf("%d", &T);
    a[i] = T * 1.0 / v;
}
scanf("%d", &T);
sort(a + 1, a + n + 1, greater<long double>());
for (i = 1; i <= n; i++)
    q[i] = q[i - 1] + a[i];
while (T--) {
    scanf("%d", &tem);
    printf("%d\n", ef(0, n, tem));
}
```

核心代码

没错，我的代码很有点难懂
将就着看看吧

THANK YOU