

CSP-J2 三校联考01 题解

T1 模法师

30pts

枚举所有合法的 y 更新答案。

100pts

对于第一种询问，要求余数最小，求最小 y ：

如果 x 不是质数，那么答案就是 x 的最小质因子；否则答案是2。

对于第二种询问，要求余数最大，求最小 y ：

尝试观察规律，在枚举的基础上，通过尝试求出前100（或前1000）的答案

```
1 //举一个例子，求x=32的模数最大值
2 int x = 32;
3 for(int i=x-1;x>=2;i--){
4     printf("x=%d,i=%d,x%%i=%d\n",x,i,x%i);
5 }
```

可以发现一个未证明的规律： $y = \lfloor \frac{x}{2} \rfloor + 1$

在这种情况下 $x \% y = (x - y) = \lceil \frac{x}{2} \rceil - 1$ 。

数学符号解释：

$\lfloor \rfloor$ 指对结果向下取整。例如 $\lfloor 2.5 \rfloor = 2$ ， $\lfloor 3.1 \rfloor = 3$

$\lceil \rceil$ 指对结果向上取整。例如 $\lceil 2.5 \rceil = 3$ ， $\lceil 3.1 \rceil = 4$

证明如下：

如果 $y' \geq \lfloor \frac{x}{2} \rfloor + 1$ ，那么 $x \% y' = x - y'$ ，当 $y' = \lfloor \frac{x}{2} \rfloor + 1$ 时取到最优；

如果 $y' \leq \lfloor \frac{x}{2} \rfloor$ ：

- 当 x 是偶数时

若 $y' = \lfloor \frac{x}{2} \rfloor$ ，那么 $x \% y' = 0$ ；

若 $y' < \lfloor \frac{x}{2} \rfloor$ ，那么由于 $x \% y' < y'$ ， $x \% y' < \lfloor \frac{x}{2} \rfloor - 1 = \lceil \frac{x}{2} \rceil - 1 = x \% y$ 。

- 当 x 是奇数时，由于 $x \% y' < y'$ ， $x \% y' < \lfloor \frac{x}{2} \rfloor = \lceil \frac{x}{2} \rceil - 1 = x \% y$ 。

那么无论如何， $y = \lfloor \frac{x}{2} \rfloor + 1$ 都是最优解。

如果你发现了一个规律，但是你却不知道该如何证明。

你可以在题目数据范围之内写一个暴力程序来初步检验规律的正确性。

T2 可持久化变量

主要理解可持久化的意思：记录变量的变化过程，并保留所有版本。

对于本题而言，需要使用数组记忆操作过程。

30pts

暴力模拟（逐步后退）

100%pts

做一个真正的“可持久化变量”，利用数组记录变化过程。然后对于back操作就可以 $O(1)$ 回退了。

1. 注意back也是一个操作。
 2. 注意回退到底的情况。
-

T3 麻将

题目大意：给定空格分割的多组字符串，判断是否满足某种组合规律。

模拟，题目怎么说，你就怎么做

如果说比较麻烦的地方，还是在字符串处理上。还好输入的行数和每行字符串数是固定的。

读入后直接判断字符串第一个位置值是数字or字母，再根据判断结果进一步判断后确定再开个数组统计牌型。

统计完成后（此时数组中已经有各种牌的数量了）直接暴力判断即可。

T4 终焉之理

题目大意：给定一长度为 n 的序列，从左到右移动依次移动长度为 k 的区间并对其排序，问几轮移动后让序列有序。

30%pts

暴力模拟题目中的操作，在每一轮中直接从左到右扫描每个区间，暴力将每个区间排序。
这个算法的时间复杂度看似是 $O(n^3 \log n)$ 的，但经过后面的可以发现其实这个算法的时间复杂度为 $O(n^2 \log n)$

50pts

我们猜测当 k 较大时，操作轮数 ans 很小。

其实我们不必在每一轮中对所有区间排序，观察每轮从左到右扫描的过程，假设当前将要排序区间的左端点为 i ，那么这次排序影响的元素在区间 $[i, i + k - 1]$ 内，它会把区间 $[i, i + k - 1]$ 的最小值固定在位置 i 上，然后右移待排序区间的时候会把第 $i + k$ 位置上的元素纳入排序的影响范围。

因此我们可以在每轮扫描的时候，先让 k 个变量直接入小根堆，用一个小根堆维护当前排序影响的元素，每次弹出最小的元素放入序列，然后选取后面第 k 个元素加入堆内。

时间复杂度 $O(ans \times n \log n)$ 。

100pts

仔细观察可以发现，每次操作后，每个元素左边的比它大的最大的 $k - 1$ 个元素会移到它的右边，那么我们用树状数组统计每个元素左边比它大的元素个数 cnt ，那么 $\max\{\left\lceil \frac{cnt}{k-1} \right\rceil\}$ 就是答案。

其实我们还有一个更加简单的做法，我们无需用树状数组统计这类元素的数量，**直接将元素排序**，若一个元素左移，它位置的偏移量就是它左边比它大的元素个数。然后套用上述公式，计算即可。这个做法的正确性是显然的，按照上面的方法计算答案同样能解决问题。

时间复杂度 $O(n \log n)$