

PDF 全国青少年信息学奥林匹克联赛模拟赛

有毒 Round

Tx_Lcy

时间：2023 年 6 月 29 日 8:45 ~ 11:45

题目名称	有毒爱排列	有毒爱 AT	妈妈生的 V
题目类型	传统型	传统型	传统型
目录	per	agc	sort
可执行文件名	per	agc	sort
输入文件名	per.in	agc.in	sort.in
输出文件名	per.out	agc.out	sort.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
子任务数目	4	6	6
子任务是否等分	否	否	否

提交源程序文件名

对于 C++ 语言	per.cpp	agc.cpp	sort.cpp
-----------	---------	---------	----------

编译选项

对于 C++ 语言	<code>-std=c++14 -O2 -lm -Wl,--stack=2147483647</code>
-----------	--

注意事项（请仔细阅读）

1. 本场比赛均为原创题，请勿外泄。
2. 由于出题人寻宝，所以题都很水，AK 的同学不要大声喧哗。
3. 本套试题官方唯一指定评测机为小机房神机，背锅人是 **syf2008**。
4. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
5. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
6. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
7. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
8. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
9. 程序可使用的栈空间大小与该题内存空间限制一致。
10. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
11. 评测在最新公布的 NOI Linux 下进行，使用 LemonLime 评测。
12. 由于某种原因，题目不一定按难度顺序排列。

有毒爱排列 (per)

【题目背景】



【题目描述】

有毒让你求长度为 n 且逆序对个数对 p 取余为 k 的排列的个数，答案对 998244353 取模。

【输入格式】

从文件 *per.in* 中读入数据。
第一行读入三个整数， n, p, k 。

【输出格式】

输出到文件 *per.out* 中。
共一行，一个整数，表示排列个数。

【样例 1 输入】

```
1 3 2 1
```

【样例 1 输出】

```
1 3
```

【样例 2 输入】

```
1 10 11 3
```

【样例 2 输出】

```
1 329980
```

【数据范围】

保证 $0 \leq k < p$ 。

子任务 1: $1 \leq n, p \leq 10$, 20 分。

子任务 2: $1 \leq n, p \leq 20$, 35 分。

子任务 3: $1 \leq n \leq 500, 1 \leq p \leq 100$, 20 分。

子任务 4: $1 \leq n \leq 2 \times 10^3, 1 \leq p \leq 100$, 25 分。

有毒爱 AT (agc)

【题目背景】

62	 mtsd	1900 (1)	400	700	800 (1)	(3)	-	-	2875	2857 → 2858 (+1)
		118:54	15:01	83:10	113:54					
63	 googol_S0	1900 (2)	400	700 (1)	800 (1)	-	-	-	2867	2706 → 2723 (+17)
		121:32	5:36	50:23	111:32					
64	 lanos212	1900 (4)	400 (1)	700	800 (3)	-	-	-	2860	2432 → 2484 (+52)
		122:22	14:25	61:45	102:22					
65	 rui_erchan	1900 (4)	400	700	800 (4)	-	-	-	2852	1200 → 1914 (+714)
		122:54	8:28	62:19	102:54					
66	 apiad	1900	400	700	800	(1)	-	-	2845	3673 → 3612 (-61)
		123:27	93:11	102:52	123:27					

有毒认为：这个 lanos 就是逊啦，agc 才切了 3 题。我的水平跟 zky 差不多，他可以切 5 题，我也能切 5 题，而且我还比他快，吊打他。

【题目描述】

有毒得到了一个字符串，这个字符串上的每一个字符都代表着某题在某场 agc 中的题目编号（例如：FEABCDD）。

我们定义吊打子序列为 ABCDE 或 ABCDEF。

有毒会询问多次，每次有毒会询问一个区间 $[l_i, r_i]$ ，设字符串 t 为 s_{l_i, \dots, r_i} 。

你需要回答 t 最多能划分成几个不相交的吊打子序列。

如不理解题意可以看样例解释。

【输入格式】

从文件 `agc.in` 中读入数据。

第一行一个整数 n ，表示字符串的长度。

接下来一行一个字符串。

第三行一个整数 q ，表示询问个数。

接下来 q 行，每行两个整数 l_i, r_i ，表示询问的区间。

【输出格式】

输出到文件 `agc.out` 中。

输出 q 行，一行一个整数，表示这次询问的答案。

【样例 1 输入】

```

1 11
2 AABBCDDDEEF
3 2
4 1 9

```

5 1 11

【样例 1 输出】1 1
2 2**【样例 1 解释】**

对于询问 1, 字符串 t 为 AABBCCDDE。下面的是一种可行的划分方案（下划黄线的表示取了）：

A A B B C C D D E

对于询问 2, 字符串 t 为 AABBCCDDEEF。下面的是一种可行的划分方案（下划黄线/蓝线的表示取了）：

A A B B C C D D E E F

【样例 2】

见选手目录下的 *agc/agc2.in* 与 *agc/agc2.ans*。

【数据范围】

保证 $1 \leq l_i \leq r_i \leq n$, 字符串里每个字母都是 A,B,C,D,E,F 中的一个。

子任务 1: $1 \leq n, q \leq 20$, 10 分。

子任务 2: $1 \leq n, q \leq 1000$, 15 分。

子任务 3: $1 \leq n \leq 5000$, $1 \leq q \leq 5 \times 10^5$, 20 分。

子任务 4: $1 \leq n, q \leq 20000$, 10 分。

子任务 5: $1 \leq n, q \leq 60000$, 25 分。

子任务 6: $1 \leq n, q \leq 5 \times 10^5$, 20 分。

妈妈生的 V (sort)

【题目背景】

机房里有一个 fishhead(别名: fishdead), 他喜欢大叫, 因为一些特殊原因, 他回去考期末考了。



他跟冯王亮有血缘关系, 但是冯王亮不是妈妈生的。

【题目描述】

鱼给了你一个 $1 \sim n$ 的排列 a 。

我们定义一次操作为执行以下两种函数之一:

- 执行一次 $f()$, 这需要 x 的代价。
- 选择一个整数 b ($1 \leq b \leq n$), 然后执行 $s(b)$, 这需要 y 的代价。

当 a 满足 $\forall i (1 \leq i \leq n), a_i = i$ 时, 停止操作, 否则继续进行下一次操作。

你需要令停止操作时花费的代价最少, 并输出这个最少代价。

如不理解题意可以看样例解释。

两种函数代码如下:

```
1 void f(){
2     for (int i=1;i<=n;++i)
3         if (a[i]<i) swap(a[i],a[a[i]]);
4 }
```

```
1 void s(int x){
2     vector<int>b;map<int,int>vis;
3     while (!vis[x]) vis[x]=1,b.push_back(x),x=a[x];
4     sort(b.begin(),b.end());
5     reverse(b.begin(),b.end());
6     for (int i=0;i<b.size();++i) a[b[i]]=b[(i+1)%b.size()];
7 }
```

【输入格式】

从文件 *sort.in* 中读入数据。

第一行三个整数 n, x, y 。

接下来一行是排列 a 。

【输出格式】

输出到文件 *sort.out* 中。

一行一个整数，最小代价。

【样例 1 输入】

```
1 6 1 1
2 6 5 3 4 1 2
```

【样例 1 输出】

```
1 2
```

【样例 1 解释】

可以执行两次 $f()$ ， $6, 5, 3, 4, 1, 2 \rightarrow 1, 2, 3, 4, 6, 5 \rightarrow 1, 2, 3, 4, 5, 6$ ，代价为 $1 + 1 = 2$ 。

也可执行 $s(1), f()$ ， $6, 5, 3, 4, 1, 2 \rightarrow 6, 1, 3, 4, 2, 5 \rightarrow 1, 2, 3, 4, 5, 6$ ，代价为 $1 + 1 = 2$ 。

【样例 2 输入】

```
1 10 2 3
2 5 8 7 6 3 1 2 9 4 10
```

【样例 2 输出】

```
1 5
```

【样例 3】

见选手目录下的 *sort/sort3.in* 与 *sort/sort3.ans*。

【样例 4】

见选手目录下的 *sort/sort4.in* 与 *sort/sort4.ans*。

【数据范围】

保证 $1 \leq x, y \leq 10^9$ 。

子任务 1: $1 \leq n \leq 5$, 15 分。

子任务 2: $1 \leq n \leq 50$, 10 分。

子任务 3: $1 \leq n \leq 5 \times 10^3$, 20 分。

子任务 4: $1 \leq n \leq 5 \times 10^4$, 15 分。

子任务 5: $1 \leq n \leq 1 \times 10^5$, 保证 $x = y$, 15 分。

子任务 6: $1 \leq n \leq 10^6$, 25 分。