

2020~2021 年高中信息学多校联合训练

模拟赛

时间：2021 年 6 月 23 日 8:00 ~ 13:00

题目名称	点灯	重构	消失的序列·改
题目类型	传统型	传统型	传统型
目录	light	refactoring	stacktwo
可执行文件名	light	refactoring	stacktwo
输入文件名	light.in	refactoring.in	stacktwo.in
输出文件名	light.out	refactoring.out	stacktwo.out
每个测试点时限	2.0 秒	1.0 秒	2.0 秒
内存限制	256 MB	256 MB	512 MB
子任务数目	3	2	5
测试点是否等分	否	否	否

提交源程序文件名

对于 C++ 语言	light.cpp	refactoring.cpp	stacktwo.cpp
-----------	-----------	-----------------	--------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项：

1. 测评时栈空间与内存限制相同。
2. 时间限制保证在标程的两倍以上，具体时限可随实际测评环境调整。
3. 函数 `main()` 的返回值类型必须是 `int`，值必须是 `0`。
4. 若无特殊说明，输入文件的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。
5. 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。
6. 测评是在每个题目对应目录下收取答案，请对每个题目建立子文件夹。
7. 题目难度与题目顺序无关，请自行选择写题顺序。
8. 题目比较简单，请独立完成，也请不要借助网络等工具。

点灯(light)

【题目描述】

小 C 打算装饰一下自己的房间，因此他买了 n 盏灯，并把它们拍成一排。

小 C 买的灯一共有 k 种颜色。如果把颜色按 $1 \sim k$ 编号，那么每盏灯的颜色都属于 $1 \sim k$ 中的一种。

小 C 把同种颜色的灯串联在了一起后，决定开始点亮这些灯（初始时灯全部处于熄灭状态）。小 C 共进行了 m 次操作，每次操作为反转第 x_i 个开关，也就是把颜色为 x_i 的灯的状态反转（若熄灭则点亮，若已点亮则熄灭）。

看着自己房间里的灯反复变化，小 C 产生了一个疑惑：

每次操作后，亮着的灯的非空连续段数量是多少？

【输入格式】

从文件 `light.in` 中读入数据。

第一行三个整数 n, k, m ，表示灯的数量，灯的颜色种类数，操作次数。

接下来一行，有 n 个整数，第 i 个整数表示第 i 盏灯的颜色 c_i 。

接下来 m 行，每行一个整数 x_i ，表示此次操作反转的开关。

【输出格式】

输出到文件 `light.out` 中。

共输出 m 行，第 i 行表示第 i 个操作后亮着的灯的非空连续段数量。

【样例输入输出】

light.in	light.out
3 2 5	2
1 2 1	1
1	1
2	0
1	1
2	
2	

【样例解释】

在第一次操作后，第一盏和第三盏灯亮着，第二盏灯熄灭，因此答案为 2。

在第二次操作后，所有灯都亮着，因此答案为 1。

在第四次操作后，没有灯亮着，因此答案为 0。

【数据范围和提示】

本题采用捆绑测试，总共 3 个子任务。

对于所有数据，满足 $1 \leq n, m \leq 2 \times 10^5, 1 \leq k \leq n, 1 \leq c_i, x_i \leq k$ 。

子任务 1 (20 pts) : 满足 $n, m \leq 1000$

子任务 2 (20 pts) : 满足 $k, m \leq 5000$

子任务 3 (60 pts) : 无特殊性质

重构(refactoring)

【题目描述】

小 Y 有一个长度为 n 的序列 a ，现在他想把这个序列重构成一个序列 b 。

小 Y 认为重构相当于把序列中的所有元素重新排列，而排列需要付出代价。对于 a 中的第 i 个元素 a_i ，如果它在 b 中被移动到了第 j 个位置，那么这个元素所花费的代价为 $|i - j|$ 。而重构整个序列的代价为所有元素的代价之和。

小 Y 想让序列 b 满足一个特殊性质： b 中的相邻两个元素奇偶性互不相同。

小 Y 想要花费最小的代价重构得到序列 b ，他需要你的帮助。

【输入格式】

从文件 `refactoring.in` 中读入数据。

第一行一个整数 n ，表示序列 a 的长度。

第二行输入 n 个整数，描述序列 a 。

【输出格式】

输出到文件 `refactoring.out` 中。

输出一行 n 个整数，表示序列 b 。

为了避免多解，如果有多个序列 b 都满足代价最小，请输出字典序最小的那个。

【样例输入输出】

refactoring.in	refactoring.out
5	5 2 1 4 3
5 3 1 4 2	

【数据范围和提示】

注意：本题中输入为序列而不是排列，序列中不保证所有元素互不相同。

本题采用捆绑测试，总共 2 个子任务。

对于 2 个子任务，均满足 $1 \leq n \leq 10^5, 1 \leq a_i \leq 10^9$ 。

子任务 1 为 35 分，且额外满足 $n \leq 10$ 。

消失的序列 · 改(stacktwo)

【题目背景】

陈题并不是过往，至少小 H 是这样认为的。

比如现在，小 H 翻起以前自己出过的题，心里总有别样的滋味：

这题的题面怎么这么繁琐...

这题的定义怎么这么不准确...

这题的英文名称是什么东西...

这题的题解怎么这么复杂...

类似如此的吐槽在小 H 心中不断涌现。尽管如此，小 H 回忆起出题时埋头苦干两个星期后灵感突现的自己，总感到欣慰。

突然之间，小 H 心中出现了一句话，这让她心头一震：

这题，好像可以加强...

【题目描述】

有些排列是可以用一个栈进行排序的，排序方式如下：

- 另外准备一个栈和一个空的序列
- 每一次操作可以把排列中的第一个数（如何排列不为空）压入栈中或把栈的栈顶元素（如果栈不为空）弹出并加入序列尾。
- 设排列长度为 n ，则最后序列中的 n 个元素应单调递增。

不难发现，并不是所有排列都可以用栈排序的。

现在，小 H 有一个长度为 n 的排列 A ，她想知道：

有多少个长度为 n 且字典序大于等于 A 的排列 p 满足 p 可用栈排序。

由于最终答案可能很大，因此你只需要输出答案对 998244353 取模后的结果。

【输入格式】

从文件 `stacktwo.in` 中读入数据。

第一行一个整数 n 。

第二行 n 个整数，描述排列 A 。

【输出格式】

输出到文件 `stacktwo.out` 中。

输出一个整数，表示答案。

【样例输入输出 1】

stacktwo.in	stacktwo.out
4 3 4 1 2	5

【样例解释 1】

满足条件的排列共 5 个，分别为：

- 4,1,2,3
- 4,1,3,2
- 4,2,1,3
- 4,3,1,2
- 4,3,2,1

【样例输入输出 2】

stacktwo.in	stacktwo.out
9 1 9 2 8 3 7 4 6 5	3761

【数据范围和提示】

本题采用捆绑测试，对于所有子任务满足 $3 \leq n \leq 10^6$ 。
保证输入为一个排列。

子任务编号	n	分值
1	≤ 9	10
2	≤ 300	20
3	≤ 3000	20
4	$\leq 10^5$	20
5	$\leq 10^6$	30