

麻烦的聚餐

[问题描述]

为了避免餐厅过分拥挤，FJ 要求奶牛们分 3 批就餐。每天晚饭前，奶牛们都会在餐厅前排队入内，按 FJ 的设想所有第 3 批就餐的奶牛排在队尾，队伍的前端由设定为第 1 批就餐的奶牛占据，中间的位置就归第 2 批就餐的奶牛了。由于奶牛们不理解 FJ 的安排，晚饭前的排队成了一个大麻烦。第 i 头奶牛有一张标明她用餐批次 $D_i(1 \leq D_i \leq 3)$ 的卡片。虽然所有 $N(1 \leq N \leq 30,000)$ 头奶牛排成了很整齐的队伍，但谁都看得出来，卡片上的号码是完全杂乱无章的。在若干次混乱的重新排队后，FJ 找到了一种简单些的方法：奶牛们不动，他沿着队伍从头到尾走一遍把那些他认为排错队的奶牛卡片上的编号改掉，最终得到一个他想要的每个组中的奶牛都站在一起的队列，例如 111222333 或者 333222111。哦，你也发现了，FJ 不反对一条前后颠倒的队列，那样他可以让所有奶牛向后转，然后按正常顺序进入餐厅。你也晓得，FJ 是个很懒的人。他想知道，如果他想达到目的，那么他最少得改多少头奶牛卡片上的编号。

所有奶牛在 FJ 改卡片编号的时候，都不会挪位置。

[输入说明]

第 1 行: 1 个整数: N

第 2~ $N+1$ 行: 第 $i+1$ 行是 1 个整数，为第 i 头奶牛的用餐批次 D_i

[输出说明]

第 1 行: 输出 1 个整数，为 FJ 最少要改几头奶牛卡片上的编号，才能让编号变成他设想中的样子

[输入样例]

```
5
1
3
2
1
1
```

[数据说明]

队列中共有 5 头奶牛，第 1 头以及最后 2 头奶牛被设定为第一批用餐，第 2 头奶牛的预设是第三批用餐，第 3 头则为第二批用餐。

[输出样例]

```
1
```

[数据说明]

如果 FJ 想把当前队列改成一个不下降序列，他至少要改 2 头奶牛的编号，一种可行的方案是：把队伍中 2 头编号不是 1 的奶牛的编号都改成 1。不过，如果 FJ 选择把第 1 头奶牛的编号改成 3 就能把奶牛们的队伍改造成一个合法的不上升序列了。