

2023 牛客 OI 赛前集训营-提高组（第一场）

比赛地址: <https://ac.nowcoder.com/acm/contest/65192>

| | | | | |
|-------------|--------------------------|--------------------------|--------------------------|---------------------------|
| 题目名称 | 情景剧 | 抽卡 | 修改 01 序列 | 虚树 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 每个测试点 时限 | C/C++1 秒, 其 他语言 2 秒 | C/C++4 秒, 其 他语言 8 秒 | C/C++1 秒, 其他 语言 2 秒 | C/C++5 秒, 其他 语言 10 秒 |
| 内存限制 | C/C++256MB, 其 他语言 2MB | C/C++256MB, 其 他语言 8MB | C/C++256MB, 其 他语言 2MB | C/C++512MB, 其 他语言 10MB |
| 子任务数目 | 20 | 20 | 10 | 20 |
| 测试点是否 等分 | 是 | 是 | 是 | 是 |

注意事项

- 所有参与牛客 OI 赛前集训营的选手必须遵守约定的纪律:
 - 比赛账号不能外传。
 - 比赛中不能抄袭代码。
 - 比赛中不能恶意卡评测。
 - 报名支付账号即为比赛账号。
 - 一旦报名牛客 OI 赛前集训营活动, 不支持退费, 请考虑清楚后报名。
 - 本活动解释权归牛客网所有, 活动介绍未尽事宜以牛客网官方解释为准。
- 欢迎关注“比赛自动姬”公众号, 关注更多比赛资讯~



情景剧

【题目描述】

白浅妹妹要在班上的 n 个同学中请一些学号连续的同学上讲台表演情景剧。用 h_i 表示学号为 i 的同学的身高，定义一个情景剧的有趣程度为上台的同学中最高的身高 \times 最矮的身高 \times 上台的人数。

白浅妹妹想知道有趣程度最大为多少。

大样例：sample.zip

【输入格式】

第一行包含一个正整数 n ，表示班级里同学的人数。第二行包含 n 个用空格隔开的正整数 h_i ，表示学号为 i 的同学身高为 h_i 。

【输出格式】

表示情景剧最大的有趣程度。

【样例 1 输入】

```
9
5 11 15 17 19 3 19 9 4
```

【样例 1 输出】

```
855
```

【说明】

让学号为 $[3,5]$ 区间内的同学表演。

【备注】

对于 30% 的数据，满足 $n \leq 10^3$

对于 60% 的数据，满足 $n \leq 10^5$

对于 80% 的数据，满足 $n \leq 10^6$

对于 100% 的数据，满足 $n \leq 2 \times 10^6, h_i \leq 10^9$

抽卡

【题目描述】

白浅妹妹最近在玩一个抽卡游戏, 现在她决定进行 n 连抽, 每次抽出的角色有一个属性 a_i , 如果相邻两次抽卡的角色属性的奇偶性不同, 白浅妹妹的郁闷值就会增加 1, 如果白浅妹妹太郁闷了, 她就会跑到你旁边嘤嘤嘤。为了不让白浅妹妹嘤嘤嘤, 你努力拿到了接下来 n 次的抽卡结果, 然后你发现其中只有 m 次抽卡的结果是固定的, 剩余 $n - m$ 次抽卡结果可指定。于是你可以调整这些可指定的抽卡结果的顺序使得白浅妹妹的郁闷值最小。然而因为游戏公司做了神奇的保密操作, 在接下来 q 段时间里, 某次结果可能由可指定变为固定, 也有可能从固定变为可指定, 你需要求出每段时间发生的变化后, 白浅妹妹的最小郁闷值是多少。sample.zip

【输入格式】

第一行包含三个整数 n, m, q 。接下来一行 n 个由空格分割的正整数 a_i , 表示接下来所有 n 次的抽卡结果的集合。

接下来 m 行每行两个正整数 p_i, b_i , 表示第 p_i 次的抽卡结果固定为 b_i 。接下来 q 行, 每行包含若干个整数, 表示一种变化, 具体如下:

+ 变化 1: 格式: $1p$ 含义: 第 p 次的抽卡结果变为不固定。

+ 变化 2: 格式: $2p$ 含义: 第 p 次的抽卡结果固定为 x 。

保证所有输入不会冲突。

【输出格式】

输出包含 q 行, 即为每次变化后白浅妹妹的最小郁闷值。

【样例 1 输入】

```
10 8 10
15 4 12 10 14 5 18 7 9 11
5 12
6 18
1 4
10 5
7 7
2 15
9 14
4 10
2 8 11
1 7
1 6
2 7 18
2 6 9
1 8
2 8 7
1 9
```

1 4

1 5

【样例 1 输出】

5

5

5

7

7

7

7

5

5

5

【备注】

| 测试点编号 | $n \leq$ | $q \leq$ | 对应样例 |
|---------|----------|----------|------|
| 1 – 3 | 20 | 5 | 2 |
| 4 – 6 | 100 | 100 | 3 |
| 7 – 10 | 5000 | 5000 | |
| 11 – 16 | 10000 | 10000 | 4 |
| 17 – 20 | 1000000 | 1000000 | 5 |

修改 01 序列

【题目描述】

长度为 n 的只包含 0 和 1 的序列，你可以对它进行多次操作。在每次操作中，你都可以选择一个数字 0 变成 1，或者选择一个数字 1 变成 0，使得最终局面满足如下特点：

对于任意相邻的两个 1，它们在序列中的距离都是 d 的倍数，给定 d ，求最小操作次数。
长度为 n 的只包含 0 和 1 的序列，每次操作选一个 0 变 1 或者 1 变 0，使得最终局面的相邻的两个 1 之间距离是 d 的倍数，问最小操作次数。

【输入格式】

第一行输入两个正整数 n, d 。

第二行输入 n 个数，表示题目所描述的序列。

【输出格式】

输出一个数，表示最小操作次数。

【样例 1 输入】

```
5 2
0 1 0 0 1
```

【样例 1 输出】

```
1
```

【说明】

将任何一个 1 变成 0，这样就没有相邻的 1 了，自然也满足题目要求。

【样例 2 输入】

```
8 2
1 0 1 0 0 0 1 1
```

【样例 2 输出】

```
1
```

【说明】

将最后一个 1 变成 0，这样序列变为 $[1,0,1,0,0,0,1,0]$ ，1 的位置分别是 $[1,3,7]$ ，其中 1 和 3 的距离是 2 的倍数，3 和 7 的距离也是 2 的倍数。

【备注】

- 对于测试点1: $1 \leq n \leq 10^5, d = 1$ 。
- 对于测试点2~3: $1 \leq n \leq 10^5, d = 2$ 。

- 对于测试点4~5: $1 \leq d \leq n \leq 10^3$ 。
- 对于测试点6~10: $1 \leq d \leq n \leq 10^5$ 。

虚树

【题目描述】

有一棵 n 个节点，边权是正整数的树，和一个 $1 \sim n$ 的排列 p 。有 Q 次询问，每次给出 l, r, k ，你需要回答在 p 的区间 $[l, r]$ 中选 k 个点，问这 k 个点构成的虚树的边权和最大是多少。sample.zip

```
inline void decode(int &l, int &r, int &k, i64 lstats, int testop) {
    lstats %= 19260817;
    if(testop) {
        l ^= lstats; l = (l % n + n) % n + 1;
        r ^= lstats; r = (r % n + n) % n + 1;
        if(l > r) std::swap(l, r);
        k ^= lstats;
        k = (k % std::min(r - l + 1, 100)) + 1;
    }
}
```

【输入格式】

第一行一个整数 id ，表示测试点编号，样例编号为 0。第二行两个整数 op, n ，表示是否强制在线和树的大小。接下来 $n - 1$ 行，每行三个正整数 u_i, v_i, w_i ，表示 u_i, v_i 之间有一条边权为 w_i 的边。接下来一行 n 个数，表示排列 p 。接下来一行一个整数 Q ，表示询问个数。接下来 Q 行，每行三个正整数 l, r, k ，含义如题。若 $op = 1$ ，则当前测试点强制在线，每次的 l, r, k 需要调用下发的解码函数。

【输出格式】

输出包含若干行，对于每个询问输出一行一个正整数表示答案。

【样例 1 输入】

```
0
0 8
2 1 168841147
3 2 185715402
4 3 225620062
5 2 229186192
6 1 56387677
7 1 912381225
8 6 897978762
6 8 4 1 3 2 5 7
10
1 4 1
3 8 4
1 3 2
```

2 8 3
 3 4 1
 1 5 5
 1 6 1
 3 7 2
 3 6 4
 1 4 3

【样例 1 输出】

0
 1721744028
 1534543050
 2446924275
 0
 1534543050
 0
 640521656
 580176611
 1534543050

【备注】

| 测试点编号 | n | Q | k | op | 特殊性质 |
|--------|--------|-------|-----|----|-------|
| 1 | 8 | 10 | 8 | 0 | 无 |
| 2 | 8 | 10 | 8 | 1 | 无 |
| 3 | 100 | 100 | 100 | 0 | |
| 4 | 100 | 100 | 100 | 1 | |
| 5 | 1000 | 10000 | 100 | 0 | |
| 6 | 1000 | 10000 | 100 | 1 | |
| 7 | 200000 | 1000 | 100 | 0 | |
| 8 | 200000 | 1000 | 100 | 1 | |
| 9 – 10 | 200000 | 10000 | 2 | 0 | |
| 11 | 200000 | 10000 | 100 | 0 | 树形态随机 |
| 12 | 200000 | 10000 | 100 | 1 | 树形态随机 |

| | | | | | |
|---------|--------|-------|-----|---|----------------------|
| 13 | 200000 | 10000 | 100 | 0 | 1 的度数为 $n - 1$ |
| 14 | 200000 | 10000 | 100 | 1 | 1 的度数为 $n - 1$ |
| 15 - 16 | 200000 | 10000 | 100 | 0 | $l = 1, r = n$ |
| 17 - 18 | 200000 | 10000 | 100 | 0 | $r - l + 1 \leq 100$ |
| 19 | 200000 | 10000 | 100 | 0 | |
| 20 | 200000 | 10000 | 100 | 1 | |