

2025 年联赛层基础能力水平测试

Round 6

时间: 2025 年 7 月 10 日 07:50 ~ 12:10

题目名称	操作	树	翻转	最小值
题目类型	传统型	传统型	传统型	传统型
目录	opt	tree	rev	min
可执行文件名	opt	tree	rev	min
输入文件名	opt.in	tree.in	rev.in	min.in
输出文件名	opt.out	tree.out	rev.out	min.out
每个测试点时限	1.0 秒	2.0 秒	2.0 秒	3.0 秒
内存限制	256 MB	512 MB	512 MB	512 MB
测试点数目	6	6	6	6
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	opt.cpp	tree.cpp	rev.cpp	min.cpp
-----------	---------	----------	---------	---------

编译选项

对于 C++ 语言	-lm -O2 -std=c++14
-----------	--------------------

注意事项与提醒 (请选手务必仔细阅读)

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. 选手提交的源程序必须存放在**已建立好的**，且**带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
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. 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**。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端中执行命令 ulimit -s unlimited 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
9. 每道题目所提交的**代码文件大小限制为 100KB**。
10. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

11. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。

12. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

13. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

14. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。15. 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。

操作 (opt)

【题目描述】

棱镜宫殿的大门前有一个奇妙的装置。这个装置上有若干个转轮，每个转轮上有一个 0 到 9 之间的数字。这些转轮从左到右排开，形成了一个正整数 N 。

小凯为了进入棱镜宫殿，他可以进行如下操作一次：

- 选择其中一些转轮转动，对于每个选择的转轮，操作之后会独立的等概率的变化成 0 到 9 之间的任意一个数字（操作后可能会出现前导零）。

如果操作后新的 N 更大了，宫殿的大门会打开，你需要帮助小凯选择一些转轮，最大化这个概率，对 998244353 取模后输出。

【输入格式】

从文件 *opt.in* 中读入数据。

一行一个正整数 N 。

【输出格式】

输出到文件 *opt.out* 中。

输出一行一个整数，表示最大的概率。

【样例 1 输入】

1 115

【样例 1 输出】

1 970293512

【样例 1 解释】

最优的操作是选择所有转轮，概率为 $8/10 + 1/10 \times 8/10 + 1/100 \times 4/10 = 884/1000$ 。

【样例 2 输入】

1 248983501

【样例 2 输出】

1 339403081

【样例 3】

见附加文件中的 `opt/ex_opt3.in` 与 `opt/ex_opt3.ans`。

【样例 4】

见附加文件中的 `opt/ex_opt4.in` 与 `opt/ex_opt4.ans`。

【样例 5】

见附加文件中的 `opt/ex_opt5.in` 与 `opt/ex_opt5.ans`。

【样例 6】

见附加文件中的 `opt/ex_opt6.in` 与 `opt/ex_opt6.ans`。

【数据范围】

对于所有数据，保证 $1 \leq N < 10^{2 \times 10^5}$ ，输入没有前导 0。

子任务 1 (10 分): $N \leq 9$ 。

子任务 2 (14 分): $N \leq 10^9$ 。

子任务 3 (11 分): 存在一个 k 满足 $N = 10^k - 1$ 。

子任务 4 (15 分): N 最多只有一个数位不为 9。

子任务 5 (23 分): N 的所有数位单调不降。

子任务 6 (27 分): 无特殊限制。

树 (tree)

【题目描述】

小凯进入了棱镜宫殿，棱镜宫殿里有 N 个房间，第 u 个房间的美丽度为 A_u 。

棱镜宫殿里还有 $N-1$ 条无向道路，联通了所有 N 个房间，也就是说这 N 个房间构成了一棵无根树。

小凯从房间 i 开始游览，如果房间 j 的美丽度大于等于 i 的美丽度，则小凯可以将房间 j 选为游览的终点。选定后小凯会沿着房间 i 到房间 j 的唯一简单路径游览，小凯想要最大化经过的边的数量。

注意这里 j 可以等于 i ，此时经过的边的数量为 0。

由于小凯不知道他会从哪个房间开始游览，所以你需要对于每个 $i(1 \leq i \leq N)$ 。求出在最优情况下小凯能经过的边的数量。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行一个整数 N 。

第二行 N 个整数 A_1, A_2, \dots, A_N 。

接下来 $N-1$ 行，每行两个整数 u_i, v_i ，表示树上的一条边。

【输出格式】

输出到文件 `tree.out` 中。

输出 N 个整数，分别表示 $i = 1, 2, \dots, N$ 的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 0 3 2 2 3
```

【样例 1 解释】

- $i = 1, j = 1, dist(i, j) = 0$
- $i = 2, j = 4, dist(i, j) = 3$
- $i = 3, j = 2, dist(i, j) = 2$
- $i = 4, j = 1, dist(i, j) = 2$
- $i = 5, j = 2, dist(i, j) = 3$

【样例 2】

见附加文件中的 tree/ex_tree1.in 与 tree/ex_tree1.ans。

【样例 3】

见附加文件中的 tree/ex_tree2.in 与 tree/ex_tree2.ans。

【样例 4】

见附加文件中的 tree/ex_tree3.in 与 tree/ex_tree3.ans。

【样例 5】

见附加文件中的 tree/ex_tree4.in 与 tree/ex_tree4.ans。

【样例 6】

见附加文件中的 tree/ex_tree5.in 与 tree/ex_tree5.ans。

【样例 7】

见附加文件中的 tree/ex_tree6.in 与 tree/ex_tree6.ans。

【数据范围】

对于所有数据, 保证 $2 \leq N \leq 2 \times 10^5, 0 \leq A_i \leq 10^9, 1 \leq u_i, v_i \leq N$, 保证构成了一棵树。

子任务 1 (13 分): $N \leq 100$ 。

子任务 2 (18 分): $N \leq 1000$ 。

子任务 3 (16 分): $u_i = i, v_i = i + 1$ 。

子任务 4 (8 分): $u_i = 1$ 。

子任务 5 (8 分): $A_i = 0$ 。

子任务 6 (37 分): 无特殊限制。

翻转 (rev)

【题目描述】

小凯获得了棱镜之力。

小凯有一个长度为 N 的整数序列 A_1, A_2, \dots, A_N ，他可以进行如下操作若干次：

- 选择序列的一个区间，对其施展棱镜之力，施展后区间内的每个数会变为它们的相反数。形式化的说，一次操作选择 $l, r (1 \leq l \leq r \leq n)$ ，然后将 A_l, A_{l+1}, \dots, A_r 乘上 -1 。

小凯想让序列 A 的每个位置都有自己的个性，所以他定义序列 A 是好的，当且仅当对于所有 $i, j (1 \leq i < j < N)$ ，有 $|A_i - A_{i+1}| \neq |A_j - A_{j+1}|$ 。

求让序列 A 变成好的所需的最小操作次数，或报告无解。

【输入格式】

从文件 `rev.in` 中读入数据。

第一行一个整数 N 。

第二行 N 个整数 A_1, A_2, \dots, A_N 。

【输出格式】

输出到文件 `rev.out` 中。

输出一个整数表示最小的操作次数，如果无解输出 -1 。

【样例 1 输入】

```
1 4
2 1 3 0 2
```

【样例 1 输出】

```
1 1
```

【样例 1 解释】

操作 $l = 2, r = 3$ ，序列变为 $\{1, -3, 0, 2\}$ ， $|A_1 - A_2| = 4, |A_2 - A_3| = 3, |A_3 - A_4| = 2$ 。

【样例 2】

见附加文件中的 `rev/ex_rev1.in` 与 `rev/ex_rev1.ans`。

【样例 3】

见附加文件中的 `rev/ex_rev2.in` 与 `rev/ex_rev2.ans`。

【样例 4】

见附加文件中的 `rev/ex_rev3.in` 与 `rev/ex_rev3.ans`。

【样例 5】

见附加文件中的 `rev/ex_rev4.in` 与 `rev/ex_rev4.ans`。

【样例 6】

见附加文件中的 `rev/ex_rev5.in` 与 `rev/ex_rev5.ans`。

【样例 7】

见附加文件中的 `rev/ex_rev6.in` 与 `rev/ex_rev6.ans`。

【数据范围】

对于所有数据，保证 $3 \leq N \leq 10^5$, $-10^9 \leq A_i \leq 10^9$ 。

子任务 1 (3 分): $N \leq 3$ 。

子任务 2 (9 分): $N \leq 9$ 。

子任务 3 (23 分): $N \leq 20$ 。

子任务 4 (23 分): $N \leq 100$ 。

子任务 5 (23 分): $N \leq 1000$ 。

子任务 6 (19 分): 无特殊限制。

最小值 (min)

【题目描述】

小凯有两个长度为 N 的整数序列 A_1, A_2, \dots, A_N 和 B_1, B_2, \dots, B_N 。

小凯现在越来越喜欢对称的东西了，当他看到两个序列 C, D 时，他会找到序列 C 的最小值 C_{\min} 和 D 的最小值 D_{\min} ，他认为这两个最小值相差越小越好，所以他定义 $f(C, D) = |C_{\min} - D_{\min}|$ 。

更进一步的，他定义 $w(l, r)$ 表示当 $C = \{A_l, A_{l+1}, \dots, A_r\}, D = \{B_l, B_{l+1}, \dots, B_r\}$ 时 $f(C, D)$ 的值，即 $\{A_l, A_{l+1}, \dots, A_r\}$ 的最小值和 $\{B_l, B_{l+1}, \dots, B_r\}$ 的最小值的差的绝对值。

对于每个 $k = 1, 2, \dots, N$ ，小凯想求出所有长度为 k 的区间 $[l, r]$ 中， $w(l, r)$ 的最小值，即求 $\min_{r-l+1=k} w(l, r)$ 。

【输入格式】

从文件 **min.in** 中读入数据。

第一行一个整数 N 。

第二行 N 个整数 A_1, A_2, \dots, A_N 。

第三行 N 个整数 B_1, B_2, \dots, B_N 。

【输出格式】

输出到文件 **min.out** 中。

输出 N 行，第 i 行一个整数表示 $k = i$ 的答案。

【样例 1 输入】

```
1 3
2 1 3 4
3 8 5 7
```

【样例 1 输出】

```
1 2
2 2
3 4
```

【样例 1 解释】

- $k = 1, l = 2, r = 2$

- $k = 2, l = 2, r = 3$
- $k = 3, l = 1, r = 3$

【样例 2】

见附加文件中的 min/ex_min1.in 与 min/ex_min1.ans。

【样例 3】

见附加文件中的 min/ex_min2.in 与 min/ex_min2.ans。

【样例 4】

见附加文件中的 min/ex_min3.in 与 min/ex_min3.ans。

【样例 5】

见附加文件中的 min/ex_min4.in 与 min/ex_min4.ans。

【样例 6】

见附加文件中的 min/ex_min5.in 与 min/ex_min5.ans。

【样例 7】

见附加文件中的 min/ex_min6.in 与 min/ex_min6.ans。

【数据范围】

对于所有数据，保证 $1 \leq N \leq 2 \times 10^5, 1 \leq A_i, B_i \leq 10^9$ 。

子任务 1 (1 分): $N \leq 100$ 。

子任务 2 (19 分): $N \leq 1000$ 。

子任务 3 (2 分): $B_i = 1$ 。

子任务 4 (33 分): $B_i = 10^9$ 。

子任务 5 (15 分): $N \leq 10^5$ 。

子任务 6 (30 分): 无特殊限制。