

全国青少年信息学奥林匹克竞赛模拟赛

NOI 2025

uyom

时间：2025 年 6 月 19 日 7:40 ~ 12:40

题目名称	树	序列	俄罗斯方块
题目类型	传统型	传统型	提交答案型
目录	tree	seq	block
可执行文件名	tree	seq	block
输入文件名	tree.in	seq.in	block.in
输出文件名	tree.out	seq.out	block.out
每个测试点时限	3.0 秒	4.0 秒	N/A
内存限制	512 MiB	512 MiB	N/A
子任务数目	10	20	5
子任务是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	tree.cpp	seq.cpp	block.cpp
-----------	----------	---------	-----------

编译选项

对于 C++ 语言	-O2 -lm
-----------	---------

注意事项（请仔细阅读）

1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
3. 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。
4. 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。
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. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 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。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

树 (tree)

【题目描述】

小 A 有一棵包含了 n 个节点的树, 定义 $f(l, r)$ 表示包含了 $[l, r]$ 中所有点的最小连通块大小, 这里连通块的大小为该连通块的点数。

小 B 有 q 个询问, 对于每次小 B 给出了两个整数 l, r ($l \leq r$), 你需要求出 $\sum_{l \leq i \leq j \leq r} f(i, j)$ 。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行一个整数 n 。

接下来 $n - 1$ 行, 每行两个整数表示了树的一条边。

第 $n + 1$ 行一个整数 q , 表示询问个数。

接下来 q 行, 每行两个整数表示了一次询问。

【输出格式】

输出到文件 `tree.out` 中。

一共 q 行, 每行一个整数, 第 i 行一个整数表示第 i 个询问的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 11
2 14
3 27
```

【样例 2】

见选手目录下的 *tree/tree2.in* 与 *tree/tree2.ans*。

【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。

【数据范围】

对于 10% 的数据满足, $n, q \leq 500$ 。

对于 20% 的数据满足, $n, q \leq 2000$ 。

对于 30% 的数据满足, $n, q \leq 7000$ 。

对于 60% 的数据满足, $n, q \leq 5 \times 10^4$ 。

另有 10% 的数据满足, $q = 1$ 。

另有 10% 的数据满足, 树是一条链。

另有 10% 的数据满足, 树的生成方式为: 先生成一个 1 到 n 的排列 p , 然后第 i 条树边为 $(p_{i+1}, p_{\lceil \frac{i}{2} \rceil})$ 。

对于 100% 的数据满足, $1 \leq n \leq 10^5, 1 \leq q \leq 5 \times 10^5, 1 \leq l \leq r \leq n$ 。

序列 (seq)

【题目描述】

我们定义一个长度为 n 的排列 P 是美丽的, 当且仅当对于所有区间 $[l, r]$ 以下两个条件有至少一个满足:

- 不存在一个整数 $k(l < k < r)$ 满足 $P_l < P_k > P_r$ 。
- 不存在一个整数 $k(l < k < r)$ 满足 $P_l > P_k < P_r$ 。

给出一个长度为 n 的随机生成的排列 p , 小 B 将重新排列这个排列 p , 使得排列 p 变得美丽, 小 B 想知道她最少需要修改的位置数量为多少。

【输入格式】

从文件 `seq.in` 中读入数据。

第一行一个整数 n 。

第二行 n 个整数 p_1, p_2, \dots, p_n 。

【输出格式】

输出到文件 `seq.out` 中。

输出一行一个整数, 表示最少需要修改的位置数。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
```

【样例 1 解释】

可以修改为 $[1, 4, 6, 5, 3, 2]$ 。

【样例 2】

见选手目录下的 `seq/seq2.in` 与 `seq/seq2.ans`。

【数据范围】

对于 100% 的数据， $1 \leq n \leq 10^4$, p 是个排列, 且随机生成。
各测试点的附加限制如下表所示：

测试点编号	$n \leq$
1	10
2	16
3	20
4	
5	500
6	
7	
8	1000
9	
10	
11	1500
12	2000
13	3000
14	4000
15	5000
16	6000
17	7000
18	8000
19	9000
20	10000

俄罗斯方块 (block)

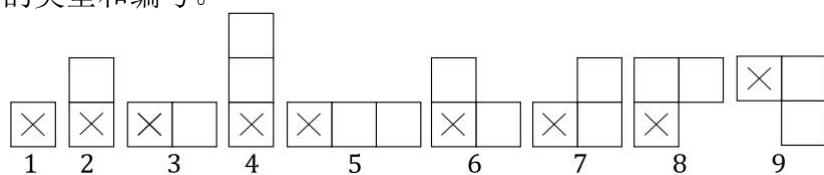
【题目描述】

俄罗斯方块是个众所周知的游戏。

在游戏中，由四个方块组成的碎片从上面掉落，游戏的目的是将碎片落在相应的位置，形成尽可能多的完全填满的行。当一行被方块完全填满时，这一行的方块消失，上面的方块掉落，从而为下落的碎片提供更多空间。

由于原游戏过于复杂，我们进行了一些简化。

在这个问题中，一共有九种碎片，每个碎片都由不超过三个方块组成，下图是所有碎片的类型和编号。



这个游戏的目标和俄罗斯方块是一样的。游戏中每个碎片将会落入一个 9×9 的区域。但与俄罗斯方块不同的是，在这个游戏中碎片不能旋转，且一旦开始下落，碎片就不能向左或向右移动。这意味着玩家只能在某个碎片开始下落的时候，指定一个列的编号，这个碎片中打 \times 的方块将会落在该列上。

每次游戏有 N 个碎片，你需要将尽可能多的碎片放入区域中，且碎片不能超过区域顶端或进行非法操作。你的得分将与成功放入的碎片数有关。

形式化地说，我们维护一个计数器，这个计数器的初值为 0，该游戏过程如下：

1. 玩家选择当前碎片最左边方块所在的列；
2. 如果选择的列合法（例如对于第 5 种碎片，选择落在第 8 列是不合法的），这个碎片会向下掉落，直到碰到障碍物为止，否则游戏结束；
3. 如果碎片完全落在区域内，则计数器的值加一，否则游戏结束；
4. 接下来检查是否有被方块完整填满的行，如果有，这一行的所有方块消失，上面的方块则在不改变相对位置的情况下向下移动；
5. 如果还有碎片，则回到第 1 步，否则游戏结束。

【输入格式】

这是一道提交答案题，共有 5 组输入数据，这些数据命名为 *block1.in* ~ *block5.in*。

输入文件第一行为一个整数 N ，代表游戏共有 N 个碎片。

接下来 N 行，第 i 行一个 $1 \sim 9$ 的整数，代表第 i 个下落的碎片种类。

输入文件保证存在一种方案，使得 N 个碎片全部落入区域中。

【输出格式】

对于每组输入，请提交对应的输出文件 *block1.out* ~ *block5.out*。

输出文件最多包含 N 行，第 i 行包含一个整数，代表第 i 个下落的碎片中，最左边方块落在的列的编号。

【样例】

下面是一个简单的游戏例子：

20 个碎片按如下顺序掉落：5, 4, 1, 6, 7, 6, 4, 4, 7, 9, 5, 5, 6, 8, 3, 4, 3, 7, 4, 2。前 17 个碎片掉落的列号如下：1, 2, 2, 4, 8, 8, 7, 4, 8, 6, 1, 1, 4, 8, 3, 7, 7，此时游戏局面如下（没有行被消除）：

						Q	Q	
		O	O			P		
L	L	L	M			P	N	N
K	K	K	M	M		P	N	I
	C		H		J	J	I	I
	B		H			J	F	
	B		H			G	F	F
	B		D			G		E
A	A	A	D	D		G	E	E
1	2	3	4	5	6	7	8	9

现在计数器的值为 17，你需要下落第 7 种碎片。

只有 2 种使得游戏不结束的掉落方案：落在第 1 列或是第 5 列。

落在第 1 列后的结果如下图：

	R					Q	Q	
R	R	O	O			P		
L	L	L	M			P	N	N
K	K	K	M	M		P	N	I
	C		H		J	J	I	I
	B		H			J	F	
	B		H			G	F	F
	B		D			G		E
A	A	A	D	D		G	E	E
1	2	3	4	5	6	7	8	9

落在第 5 列后的结果如下图（有一行被填满后消失）：

						Q	Q	
		O	O		R	P		
K	K	K	M	M		P	N	I
	C		H		J	J	I	I
	B		H			J	F	
	B		H			G	F	F
	B		D			G		E
A	A	A	D	D		G	E	E
1	2	3	4	5	6	7	8	9

【评分方式】

设你在测试点游戏结束时的计数器的值为 X ，则该测试点的得分为：

$$15 \times \frac{\min(2 \times X, N)}{N} + 5 \times \frac{\max(2 \times X - N, 0)}{N}。$$

【提示】

下发了一个可视化工具 *visual.html*，选手可自行根据需要使用。