

清华推研机试复刻组

THUPOST REMAKE

清华考研机试 2025

时间：2025 年 3 月 24 日 18:00 ~ 22:00

题目名称	见	Bloxorz	任务分配
题目类型	传统型	传统型	传统型
输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB
子任务数目	8	4	6
测试点是否等分	否	否	否

见 (See)

【题目描述】

给一个 $n \times n$ 的网格图，从上到下第 i 行，从左到右第 j 列有一个高度为 $h_{i,j}$ 的柱子。

定义 l_i 为：从第 i 行的左侧看去，能看到多少根柱子；

- 如果一个更远的柱子的高度 不超过 一个更近的柱子的高度，那么其不会被看到。

定义 r_i 为：从第 i 行的右侧看去，能看到多少根柱子；

定义 u_i 为：从第 i 列的上侧看去，能看到多少根柱子；

定义 d_i 为：从第 i 列的下侧看去，能看到多少根柱子；

请对每个 $i \in [1, n]$ ，求出 l_i, r_i, u_i, d_i 。

【输入格式】

从标准输入读入数据。

第一行，一个正整数 n ；

接下来 n 行，每行 n 个正整数，其中第 i 行第 j 列为 $h_{i,j}$ 。

【输出格式】

输出到标准输出。

n 行，每行四个正整数 l_i, r_i, u_i, d_i ，意义如上所述。

【样例 1 输入】

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

【样例 1 输出】

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

【样例 1 解释】

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

【样例 2】

见题目目录下的 *2.in* 与 *2.ans*。

【子任务】

本题采用捆绑测试，你只有通过一个子任务中的所有测试点才能得到该子任务的分数。

对于 10% 的数据， $n = 1$ ；

对于 20% 的数据， $n \leq 2$ ；

对于 30% 的数据， $n \leq 3$ ；

对于 40% 的数据， $n \leq 4$ ；

对于 50% 的数据， $n \leq 5$ ；

对于 60% 的数据， $n \leq 6$ ；

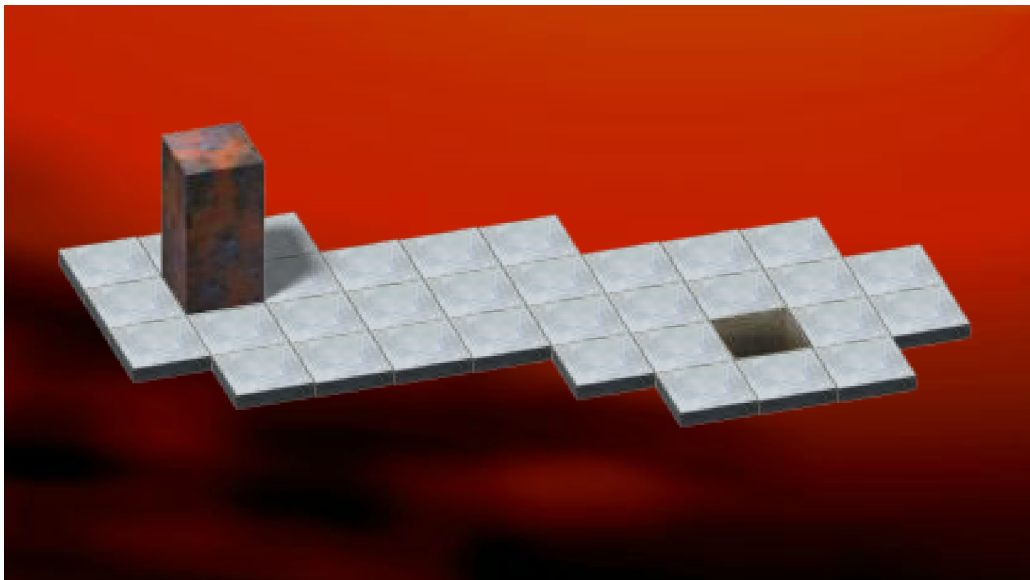
对于 80% 的数据， $n \leq 20$ ；

对于 100% 的数据， $1 \leq n \leq 100$, $1 \leq h_{i,j} \leq n$ 。

Bloxorz (Bloxorz)

【题目描述】

Bloxorz 是一款风靡全球的益智游戏，以其独特的方块操控和复杂的关卡设计而著称。在这个游戏中，玩家需要使用键盘方向键来控制方块的移动，使其到达指定的终点。



具体的游戏规则如下：给定一个平面网格区域，每一格为空地，地面或玻璃。在地面上放置着一块 $1 \times 1 \times 2$ 的长方体木块，可以竖放在一格地面上，或平放在相邻的两格地面、或相邻的两格玻璃、或相邻的一格地面和一格玻璃上。每次操作可以选择四个方向中的一个，将木块向该方向滚动 90。游戏的目标是以最少的移动次数将盒子竖放在唯一的目標地砖上。初始位置和目标位置的区域类型均为地面，而非玻璃或者空地。

给定一个 $n \times m$ 的平面网格，其中 # 表示空地，. 表示地面，E 表示玻璃，X 表示初始位置，O 表示目标位置，求达成游戏目标的最小操作数。

【输入格式】

从标准输入读入数据。
本题包含多组测试数据。
输入的第一行包含一个正整数 T ，表示测试数据组数。
对于每组测试数据：
输入的第一行包含两个正整数 n, m ，表示平面网格的大小。
输入的第 $i + 1$ ($1 \leq i \leq n$) 行包含一个长度为 m 的仅由 #、.、E、X、O 组成的字符串，描述给定的游戏局面，具体含义如【题目描述】中所示。

【输出格式】

输出到标准输出。

对于每组测试数据：

输出一行一个整数表示到达成游戏目标的最小操作数。特别地，若游戏目标无法达成，则输出 -1 。

【样例 1 输入】

```
1 1
2 7 7
3 #####
4 #..X###
5 #...##O#
6 #....E#
7 #....E#
8 #.....#
9 #####
```

【样例 1 输出】

```
1 10
```

【子任务】

对于所有测试数据，保证 $1 \leq T \leq 5$, $3 \leq n, m \leq 500$ ，保证每个游戏局面都有恰好一个 O 与恰好一个 X 或相邻的两个 X 。保证给定的游戏局面中的第一行、第一列、最后一行、最后一列均为 $\#$ 。

本题采用捆绑测试，你只有通过一个子任务中的所有测试点才能得到该子任务的分数。

子任务编号	分值	$n, m \leq$
1	30	20
2	20	50
3	20	300
4	30	500

任务分配 (Task)

【题目描述】

给定一棵 n 个点的有根树，其中节点 1 为根，节点 i ($2 \leq i \leq n$) 的父亲为 f_i 。

有 n 个任务，第 i 个任务的耗时为 t_i 。你需要将这 n 个任务分配给 n 个节点，满足每个节点恰好完成一个任务。于此同时，每个节点均有一个限制，节点 i ($1 \leq i \leq n$) 的限制为 w_i ，要求在节点 i 的子树内的所有任务的耗时均不能超过 w_i 。

现在，你可以选择一个节点，并将其的限制增加 k ，其中 k 为任意非负整数，你需要求出最小的 k ，使得在增大一个节点的限制后，存在一种任务分配的方案满足所有限制，保证存在至少一种增加限制的方案使得存在一种任务分配的方案满足所有限制。

【输入格式】

从标准输入读入数据。

输入的第一行包含一个正整数 n ，表示树的点数与任务的个数。

输入的第二行包含 $n - 1$ 个正整数 f_2, \dots, f_n ，表示 $2 \sim n$ 号节点的父亲。

输入的第三行包含 n 个正整数 t_1, t_2, \dots, t_n ，表示每个任务的耗时。

输入的第四行包含 n 个正整数 w_1, w_2, \dots, w_n ，表示每个节点的限制。

【输出格式】

输出到标准输出。

输出一行一个非负整数 k ，表示限制增量的最小值。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
```

【样例 1 解释】

一种可行的方案为：将节点 2 的限制增加 2 后，将耗时为 6, 3, 2, 2, 5, 1 的任务依次分配给节点 1 ~ 6。

【子任务】

对于所有测试数据，保证 $1 \leq n \leq 10^5$, $1 \leq f_i < i$, $1 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq n$, $1 \leq w_i \leq n$ 。保证存在至少一种增加限制的方案使得存在一种任务分配的方案满足所有限制。

对于额外子任务 7 和 8，保证 $1 \leq n \leq 3 \times 10^6$, $1 \leq f_i < i$, $1 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq n$, $1 \leq w_i \leq n$ 。保证存在至少一种增加限制的方案使得存在一种任务分配的方案满足所有限制。

本题采用捆绑测试，你只有通过一个子任务中的所有测试点才能得到该子任务的分数。

子任务编号	分值	$n \leq$	特殊性质
1	15	6	无
2	15	100	无
3	15	10^3	无
4	15	10^5	$f_i = i - 1$
5	15	10^5	$f_i = 1$
6	25	10^5	无