

SEGTREE2: Segment Tree

题目描述

线段树是一种可以高效处理序列问题的数据结构。

通常，我们使用下面的过程构建一棵线段树：

```
createTree(left, right):
    currentNode = new node()
    currentNode->left = left
    currentNode->right = right
    currentNode->tag = 0
    if (left < right):
        middle = (left + right) / 2
        currentNode->leftSon = createTree(left, middle)
        currentNode->rightSon = createTree(middle + 1, right)
    return currentNode
```

其调用方法为：

```
root = init(1, N)
```

考虑下面的线段树操作：

```
changeSegtree(currentNode, left, right):
    currentNode->tag = 1

    if (left == currentNode->left) and (right == currentNode->right):
        return

    newRight = min(right, currentNode->leftSon->right)
    if (left <= newRight):
        changeSegtree(currentNode->leftSon, left, newRight)

    newLeft = max(left, currentNode->rightSon->left)
    if (newLeft <= right):
        changeSegtree(currentNode->rightSon, newLeft, right)

change(left, right):
    changeSegtree(root, left, right)
```

最后，下面的过程描述了在线段树上的中序遍历：

```
traverse(currentNode):
    if (currentNode->leftSon != NULL):
        traverse(currentNode->leftSon)
    print currentNode->tag
    if (currentNode->rightSon != NULL):
        traverse(currentNode->rightSon)

traverse(root)
```

给出 N 的值（即构建线段树时输入的序列的长度，也即 `init` 方法的第二个参数），以及 `traverse` 方法的运行结果。你的任务是求出满足下面条件的最少的不相交区间数量，或者指出不存在满足条件的方案。条件要求，在一棵新构造的线段树上，对于这些区间执行 `change` 方法后，`traverse` 方法会给出与输入相同的结果。

输入格式

输入的第一行包含一个整数 T ，代表测试数据的组数。接下来是 T 组数据。

每组数据的第一行包含一个整数 N ，代表线段树的范围。

第二行包含 M 个空格分隔的整数 T_1, T_2, \dots, T_M ，为按照中序遍历顺序给出的线段树各节点的标记，即 `traverse` 方法的运行结果。此处 M 是线段树的节点个数，你需要自己计算。

输出格式

对于每个询问，输出一行，包含一个整数，代表需要执行 `change` 方法的最少的区间数量。如果无法得到给定的标记序列，那么输出 -1 。

数据范围

- $1 \leq T \leq 5 \cdot 10^4$
- $1 \leq N \leq 10^5$
- $1 \leq \text{输入中所有 } N \text{ 之和} \leq 5 \cdot 10^5$

样例数据

输入	输出
2	2
2	-1
1 1 1	
2	
1 0 1	

样例解释

在第一个询问中，我们可以调用 `change(1, 1)` 和 `change(2, 2)`。

在第二个询问中，无法得到给定的标记序列。

时间限制

1 秒

Problem Setters: Hasan Jaddouh, Kevin Charles Atienza, Sergey Kulik, Utkarsh Lath, Vasia Antoniuk

Problem Testers: Kevin Charles Atienza, Sergey Kulik, Utkarsh Lath, Vasia Antoniuk

Translated by: Hu Zecong