

SEGTREE2: Segment Tree

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контексте.

Условие:

Дерево отрезков — очень мощная структура данных, которая позволяет эффективно выполнять некоторые запросы на массиве.

Обычно дерево отрезков создается следующим образом:

```
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
```

Вначале нужно вызвать `root = createTree(1, N)`

Давайте рассмотрим следующую операцию с деревом:

```
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)
```

CODECHEF

А также обход дерева:

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

traverse(root)
```

Вам дано число N — размер массива (второй параметр функции **init**) и список чисел, которые выведет функция обхода дерева. Вам нужно найти такую последовательность попарно непересекающихся отрезков, что после вызова функции **change** для этих отрезков, обход дерева выведет такие же числа, что и даны на входе.

Два отрезка (l_1, r_1) и (l_2, r_2) пересекаются, если существует такое i , что $l_1 \leq i \leq r_1$ и $l_2 \leq i \leq r_2$.

Формат ввода:

Первая строка содержит целое число T , обозначающее количество тестовых случаев.

Первая строка каждого тестового случая содержит ровно одно целое число N — размер массива.

Вторая строка каждого тестового случая содержит M разделенных пробелами целых чисел — результат работы функции обхода дерева. Число M (количество вершин дерева отрезков) вы должны посчитать сами.

Формат вывода:

Для каждого тестового случая вывести одну строку, содержащую минимальное количество непересекающихся отрезков или -1, если не существует такого множества отрезков.

Ограничения:

- $1 \leq T \leq 5 \times 10^4$
- $1 \leq N \leq 10^5$
- $1 \leq \text{Сумма всех } N \text{ в каждом тестовом файле} \leq 5 \times 10^5$

CODECHEF

Примеры тестов:

Входные данные:

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

Выходные данные:

```
2
-1
```

Пояснение:

Тестовый случай 1: Можно вызвать `change(1, 1)` и `change(2, 2)`.

Тестовый случай 2: Не существует такой последовательности вызовов `change()`, которая приведет до необходимого вывода функции обхода дерева.