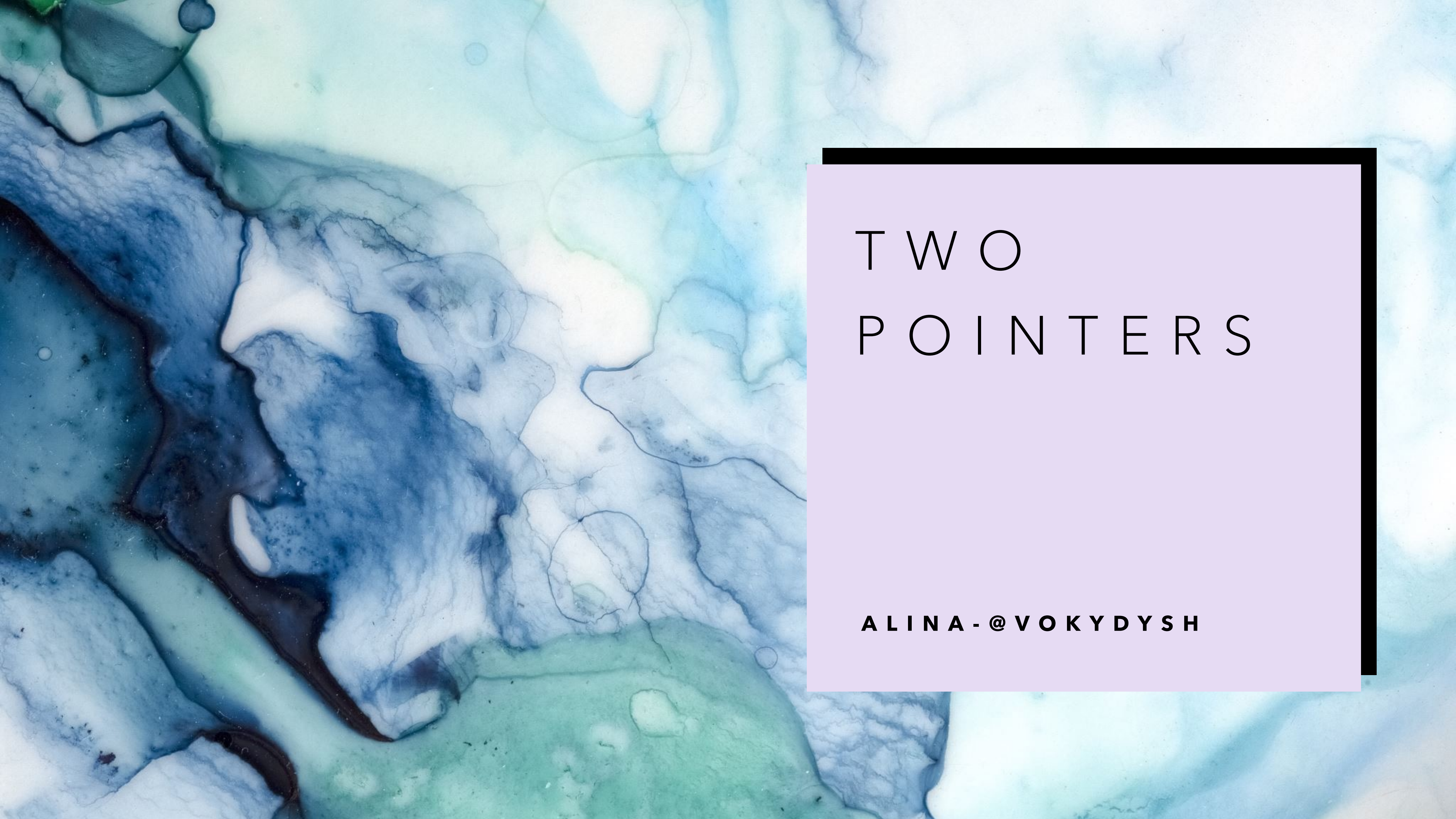
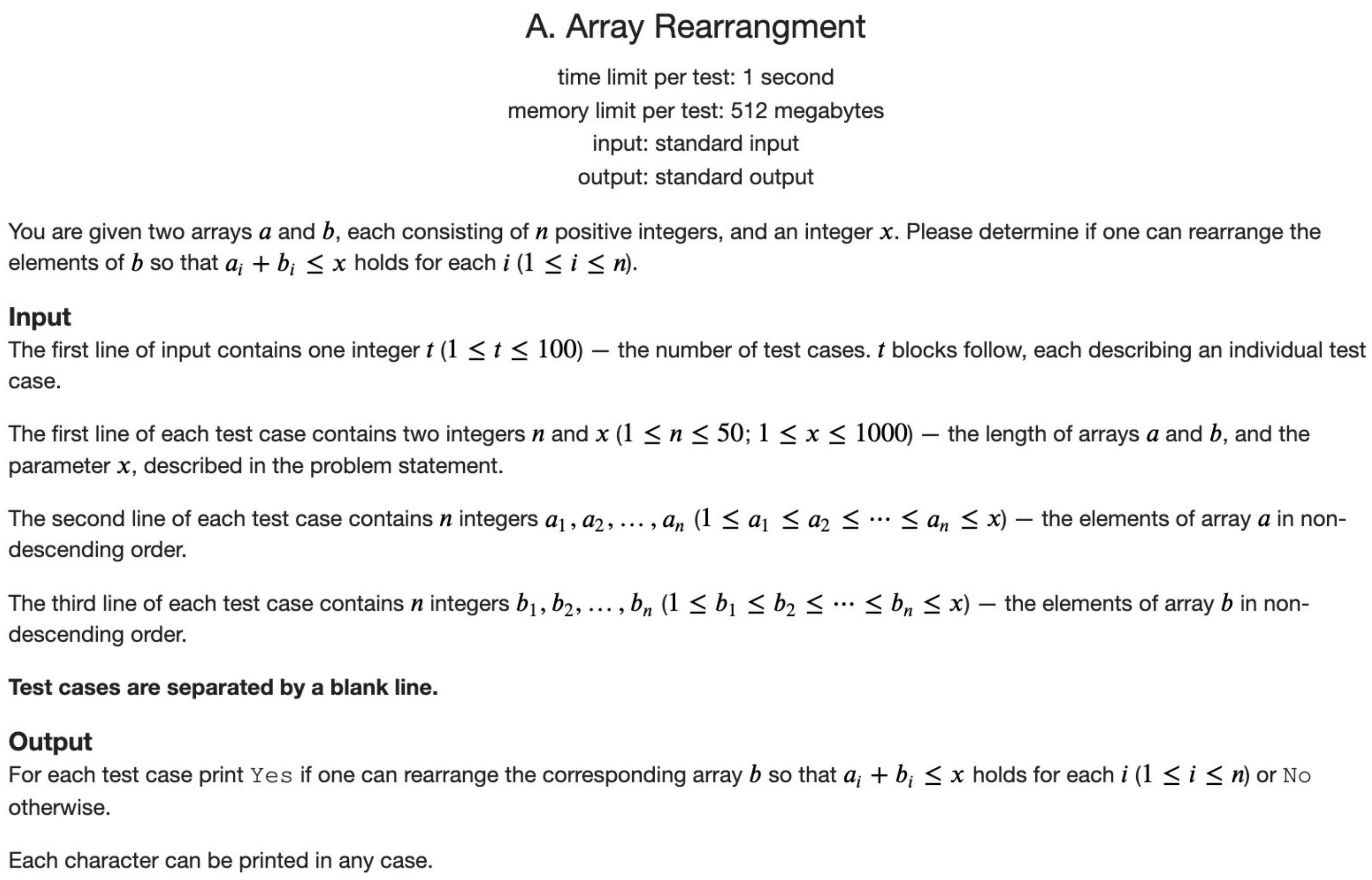


TWO POINTERS

ALINA-@VOKYDYSH

PROBLEMA



A. Array Rearrangment

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given two arrays a and b, each consisting of n positive integers, and an integer x. Please determine if one can rearrange the elements of b so that $a_i + b_i \leq x$ holds for each i ($1 \leq i \leq n$).

Input

The first line of input contains one integer t ($1 \leq t \leq 100$) — the number of test cases. t blocks follow, each describing an individual test case.

The first line of each test case contains two integers n and x ($1 \leq n \leq 50$; $1 \leq x \leq 1000$) — the length of arrays a and b, and the parameter x, described in the problem statement.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_1 \leq a_2 \leq \cdots \leq a_n \leq x$) — the elements of array a in non-descending order.

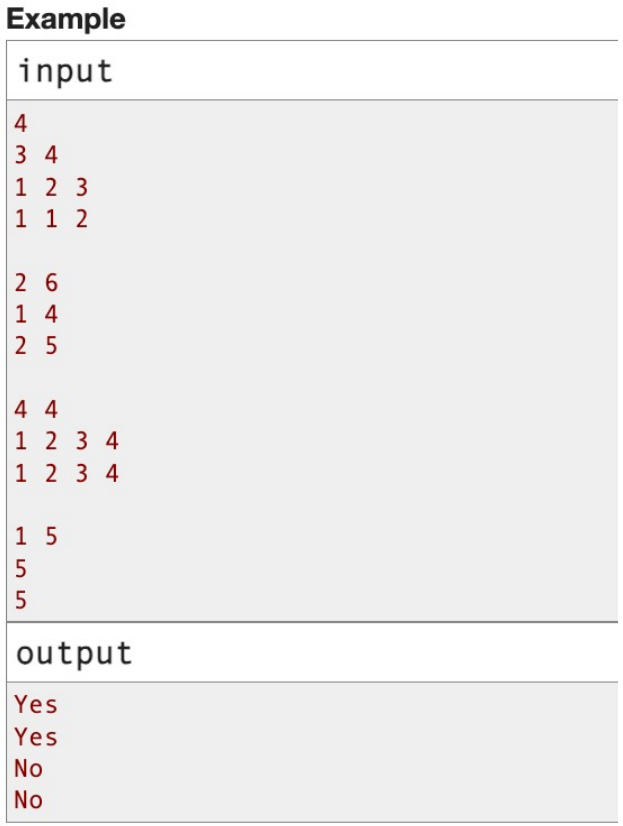
The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_1 \leq b_2 \leq \cdots \leq b_n \leq x$) — the elements of array b in non-descending order.

Test cases are separated by a blank line.

Output

For each test case print `Yes` if one can rearrange the corresponding array b so that $a_i + b_i \leq x$ holds for each i ($1 \leq i \leq n$) or `No` otherwise.

Each character can be printed in any case.



Example

input

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

2 6
1 4
2 5

4 4
1 2 3 4
1 2 3 4

1 5
5
5
```

output

```
Yes
Yes
No
No
```

Arreglos de n enteros

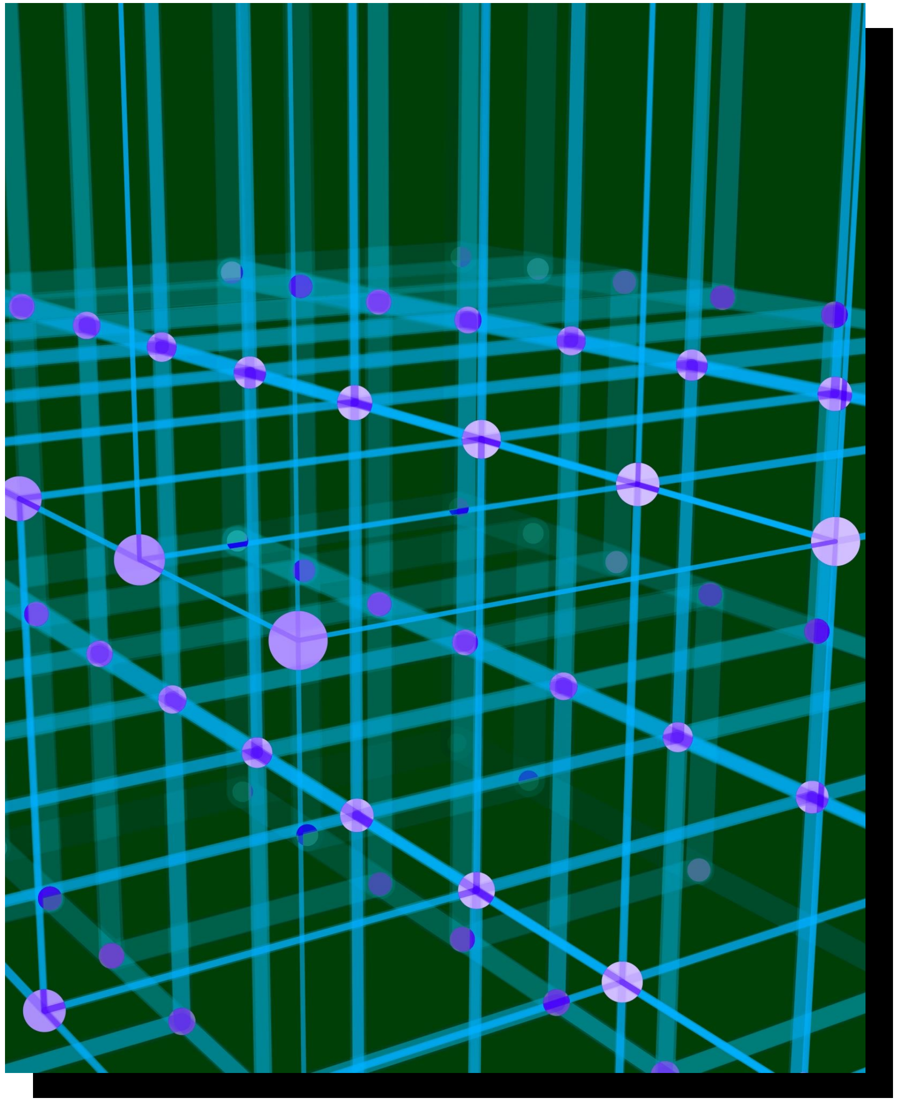
$x=9$ $x=10$

1	3	5	7	8

2	3	4	5	7

¿COMPLEJIDAD?

O(   )



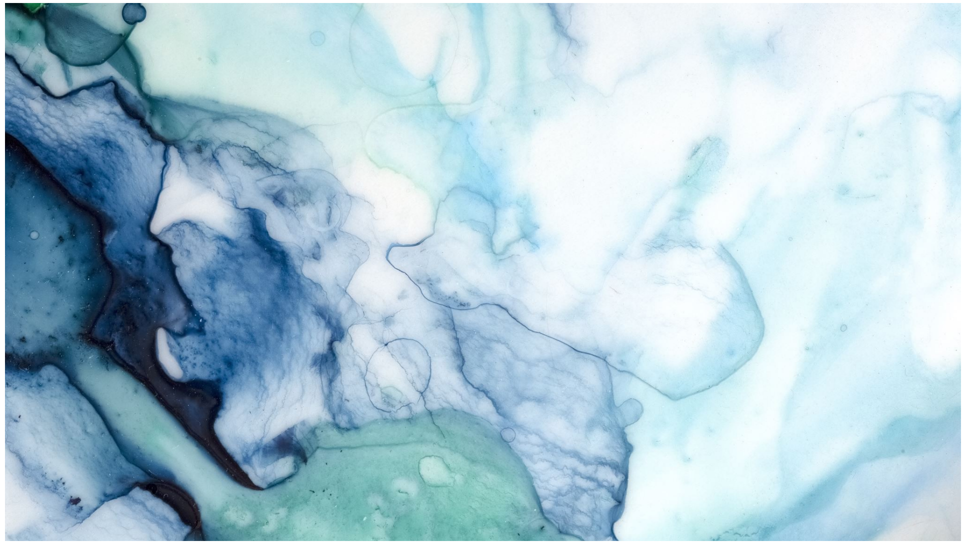
¿Qué es two pointers?

Uso de dos indicadores para manipular un arreglo o cadena.

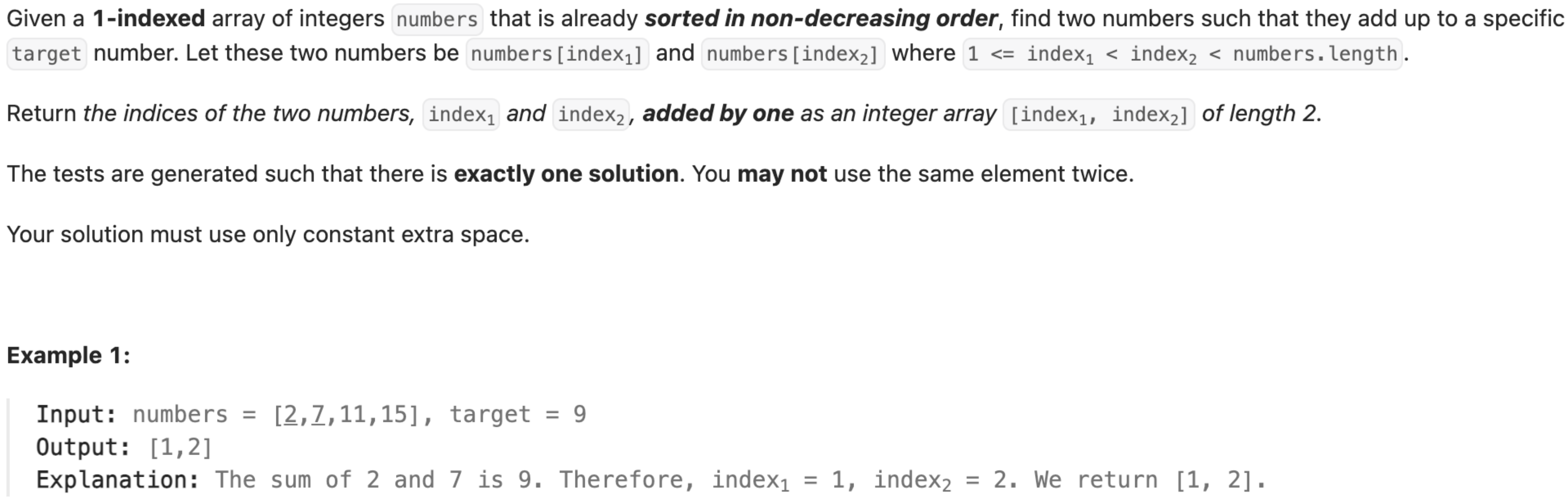
- Pointer: Indicador de posición, variables que guardan índices.
- Al utilizer dos indicadores a la vez, podemos disminuir significativamente la cantidad de operaciones necesarias para resolver un problema.

¿CUÁNDO UTILIZAR TWO POINTERS?

Cuando queremos comparer objetos con otros objetos dentro de una misma estructura.



PROBLEMA



Given a **1-indexed** array of integers `numbers` that is already ***sorted in non-decreasing order***, find two numbers such that they add up to a specific `target` number. Let these two numbers be numbers[index_1] and numbers[index_2] where $1 <= \text{index}_1 < \text{index}_2 < \text{numbers.length}$.

Return *the indices of the two numbers,* index_1 *and* index_2*,* ***added by one*** *as an integer array* [index_1, index_2] *of length 2.*

The tests are generated such that there is **exactly one solution**. You **may not** use the same element twice.

Your solution must use only constant extra space.

Example 1:

> **Input:** numbers = [2,7,11,15], target = 9
> **Output:** [1,2]
> **Explanation:** The sum of 2 and 7 is 9. Therefore, index_1 = 1, index_2 = 2. We return [1, 2].

IDEAS DE SOLUCIÓN

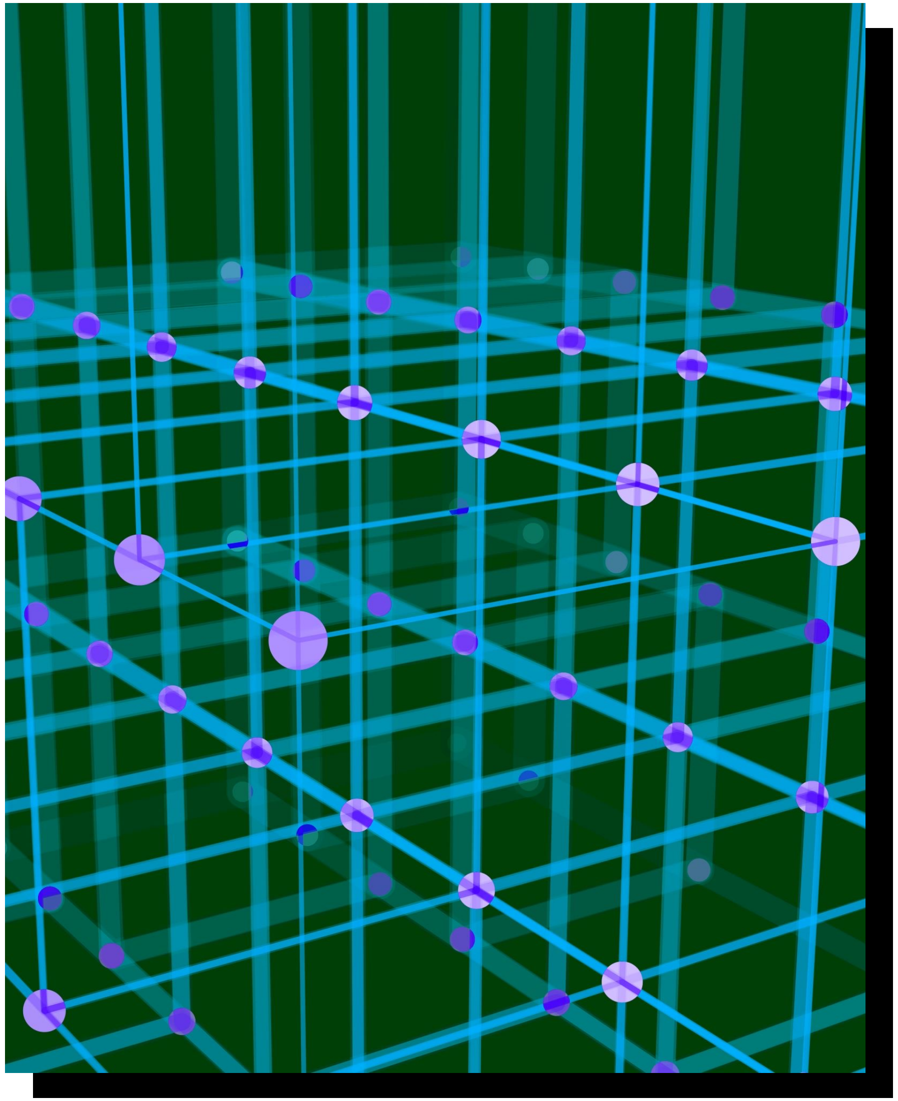
target=12

1	3	5	7	8

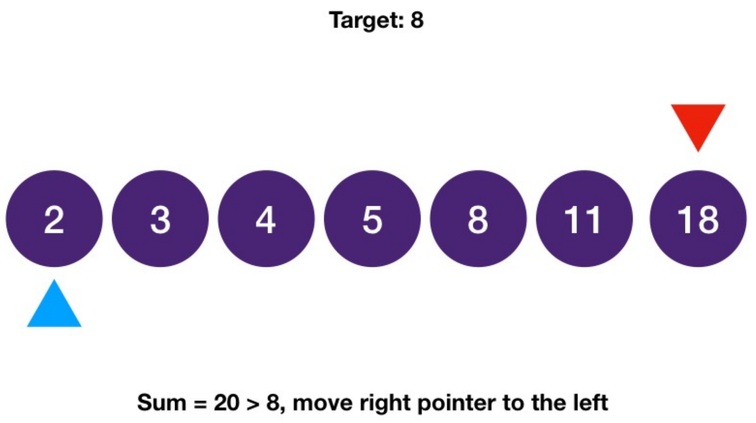
¿COMPLEJIDAD?

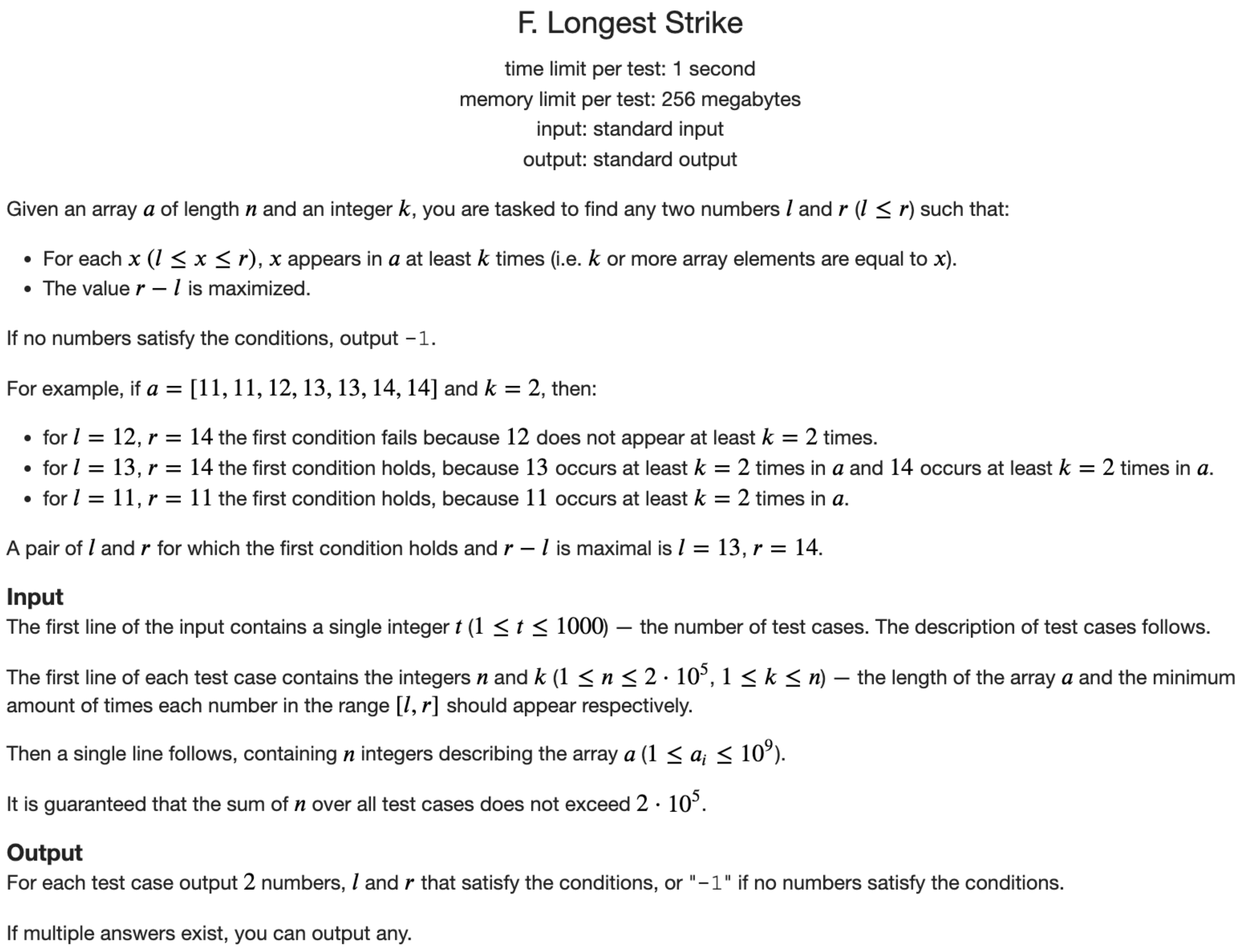
O(   )

O(   )



GRACIAS POR SU ATENCIÓN





F. Longest Strike

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Given an array a of length n and an integer k, you are tasked to find any two numbers l and r ($l \le r$) such that:

- For each x ($l \le x \le r$), x appears in a at least k times (i.e. k or more array elements are equal to x).
- The value $r - l$ is maximized.

If no numbers satisfy the conditions, output `-1`.

For example, if $a = [11, 11, 12, 13, 13, 14, 14]$ and $k = 2$, then:

- for $l = 12, r = 14$ the first condition fails because 12 does not appear at least $k = 2$ times.
- for $l = 13, r = 14$ the first condition holds, because 13 occurs at least $k = 2$ times in a and 14 occurs at least $k = 2$ times in a.
- for $l = 11, r = 11$ the first condition holds, because 11 occurs at least $k = 2$ times in a.

A pair of l and r for which the first condition holds and $r - l$ is maximal is $l = 13, r = 14$.

Input

The first line of the input contains a single integer t ($1 \le t \le 1000$) — the number of test cases. The description of test cases follows.

The first line of each test case contains the integers n and k ($1 \le n \le 2 \cdot 10^5$, $1 \le k \le n$) — the length of the array a and the minimum amount of times each number in the range $[l, r]$ should appear respectively.

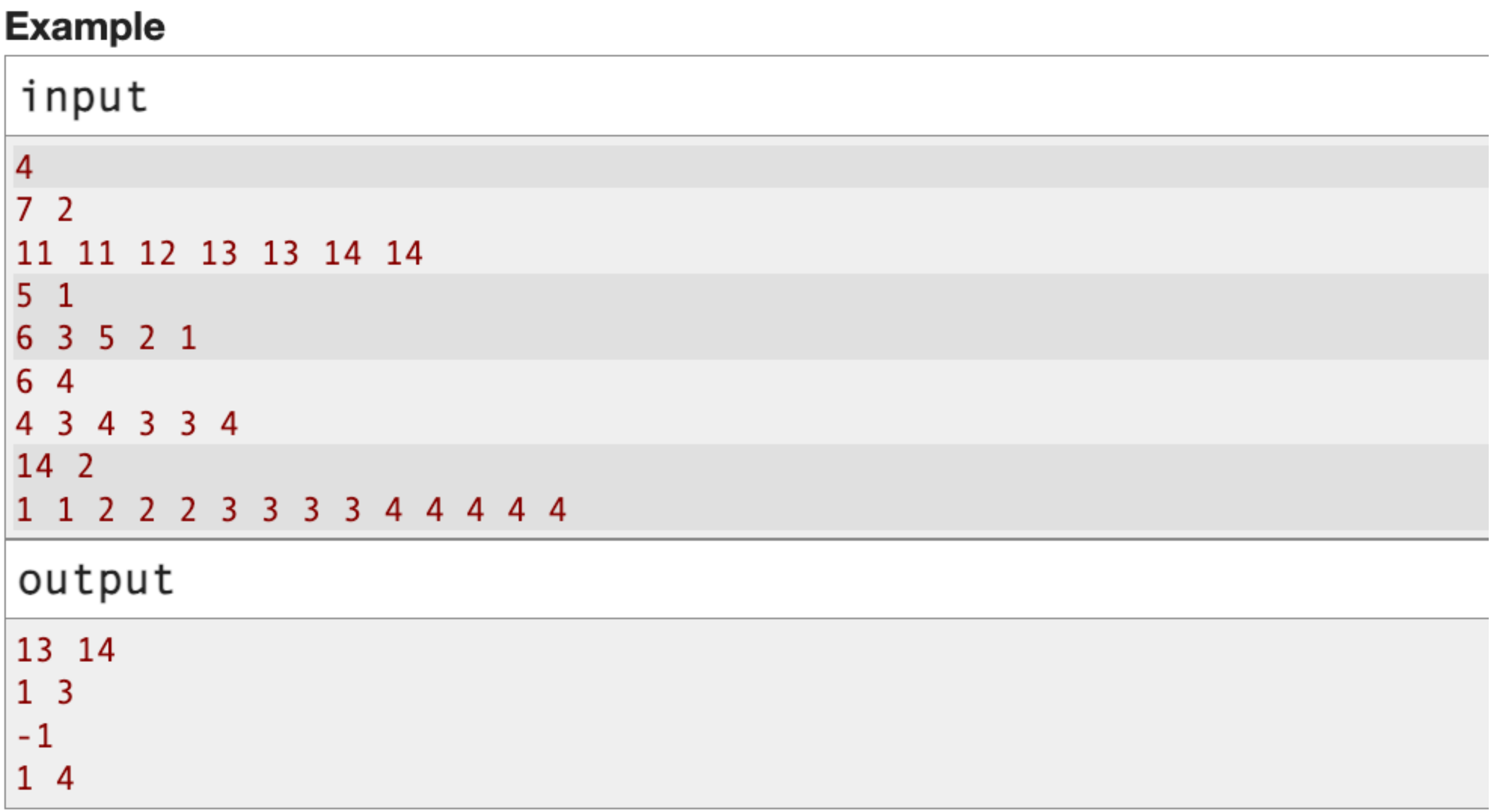
Then a single line follows, containing n integers describing the array a ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output 2 numbers, l and r that satisfy the conditions, or "`-1`" if no numbers satisfy the conditions.

If multiple answers exist, you can output any.



Example

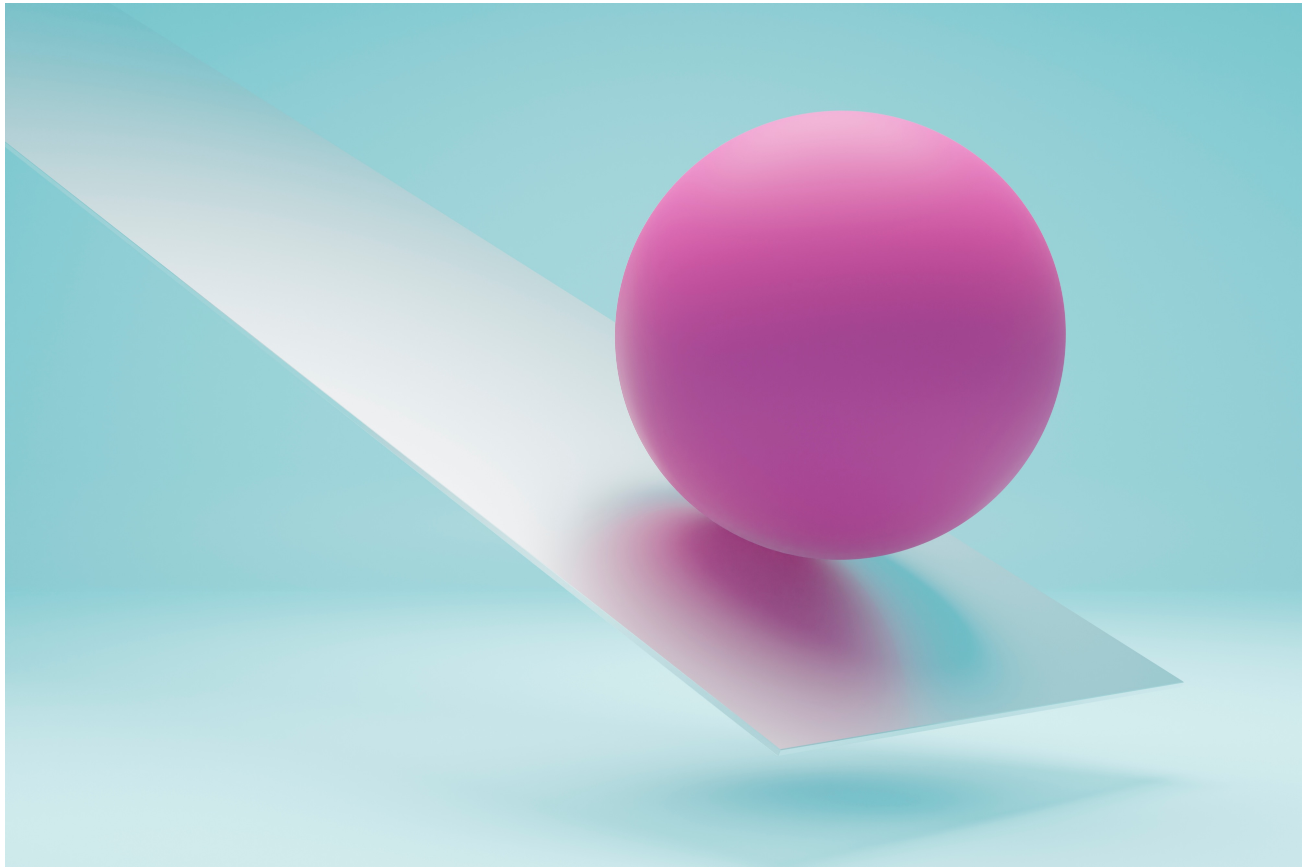
input

```
4
7 2
11 11 12 13 13 14 14
5 1
6 3 5 2 1
6 4
4 3 4 3 3 4
14 2
1 1 2 2 2 3 3 3 3 4 4 4 4 4
```

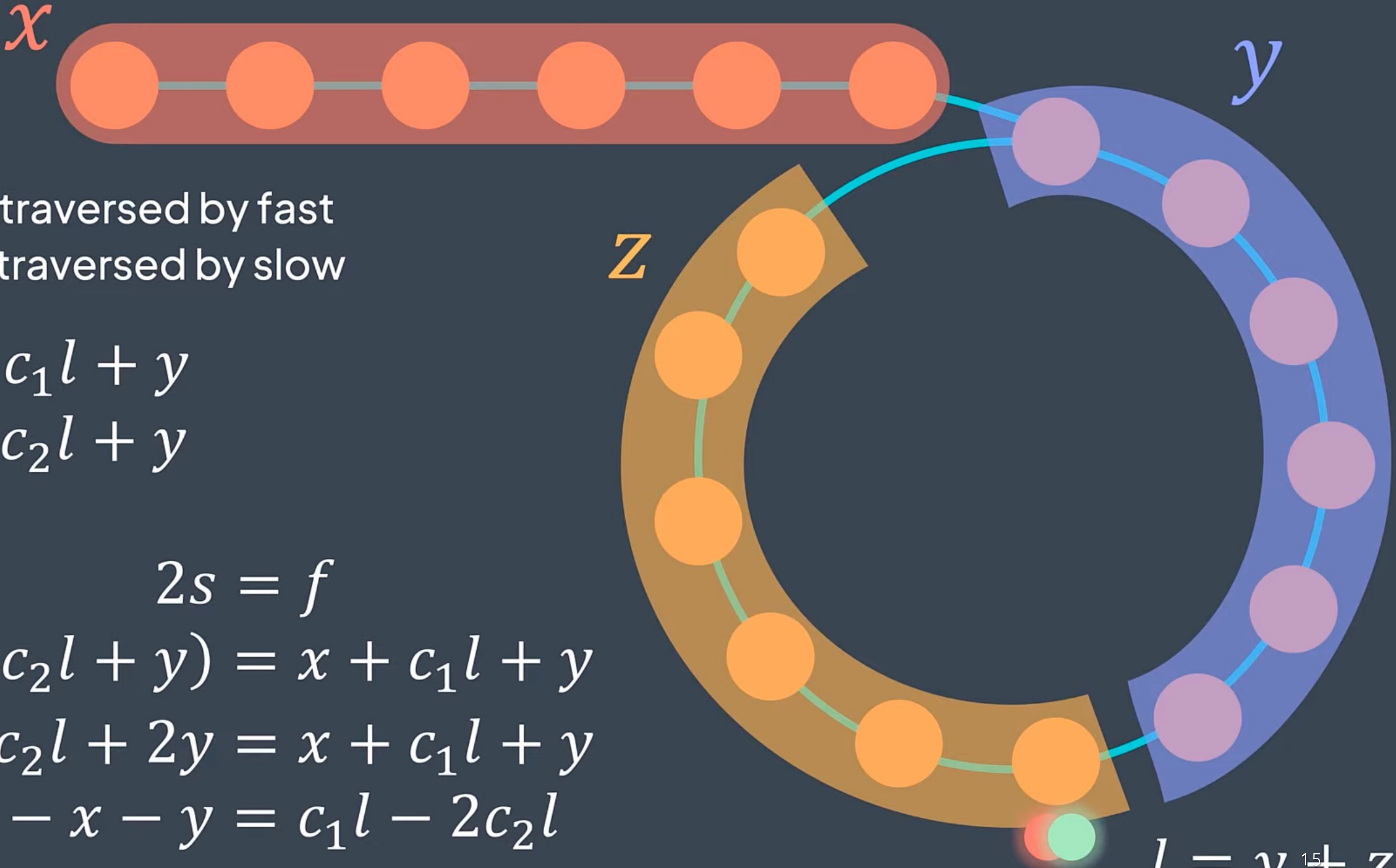
output

```
13 14
1 3
-1
1 4
```

DETECCIÓN DE CICLOS



TORTUGA Y LIEBRE



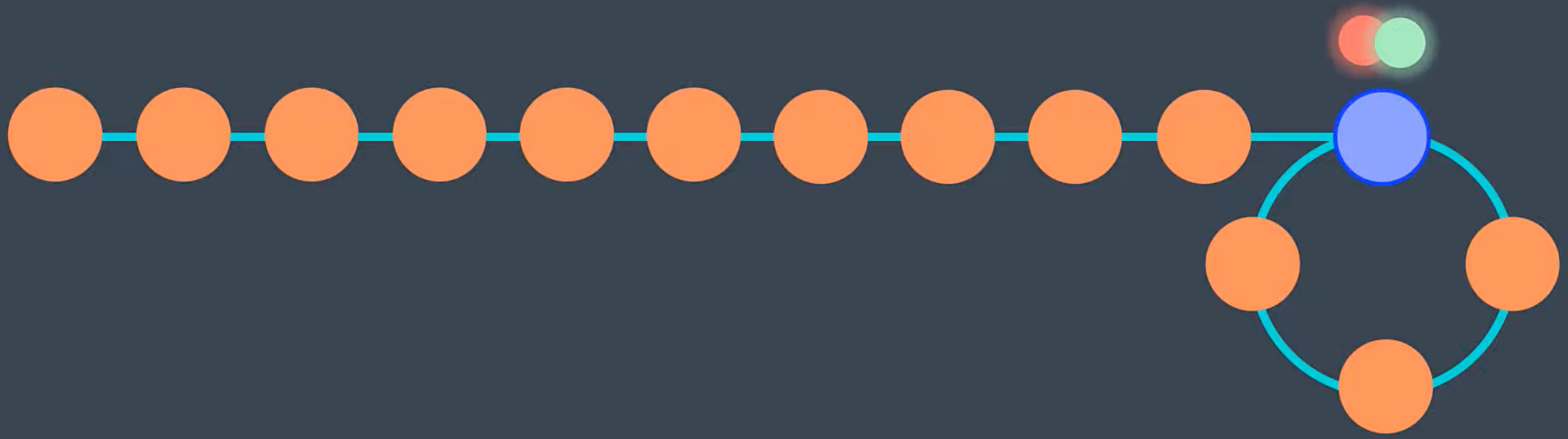
f: distance traversed by fast

s: distance traversed by slow

$$f = x + c_1 l + y$$
$$s = x + c_2 l + y$$

$$2s = f$$
$$2(x + c_2 l + y) = x + c_1 l + y$$
$$2x + 2c_2 l + 2y = x + c_1 l + y$$
$$2x + 2y - x - y = c_1 l - 2c_2 l$$

$$l = y + z$$



$$x = c_3 l + z$$

141. Linked List Cycle

Easy 👍 14.3K 👎 1.2K ☆

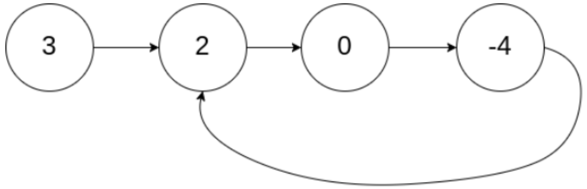
🔒 Companies

Given `head`, the head of a linked list, determine if the linked list has a cycle in it.

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the `next` pointer. Internally, `pos` is used to denote the index of the node that tail's `next` pointer is connected to. **Note that `pos` is not passed as a parameter**.

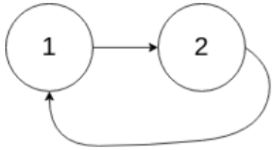
Return `true` *if there is a cycle in the linked list*. Otherwise, return `false`.

Example 1:

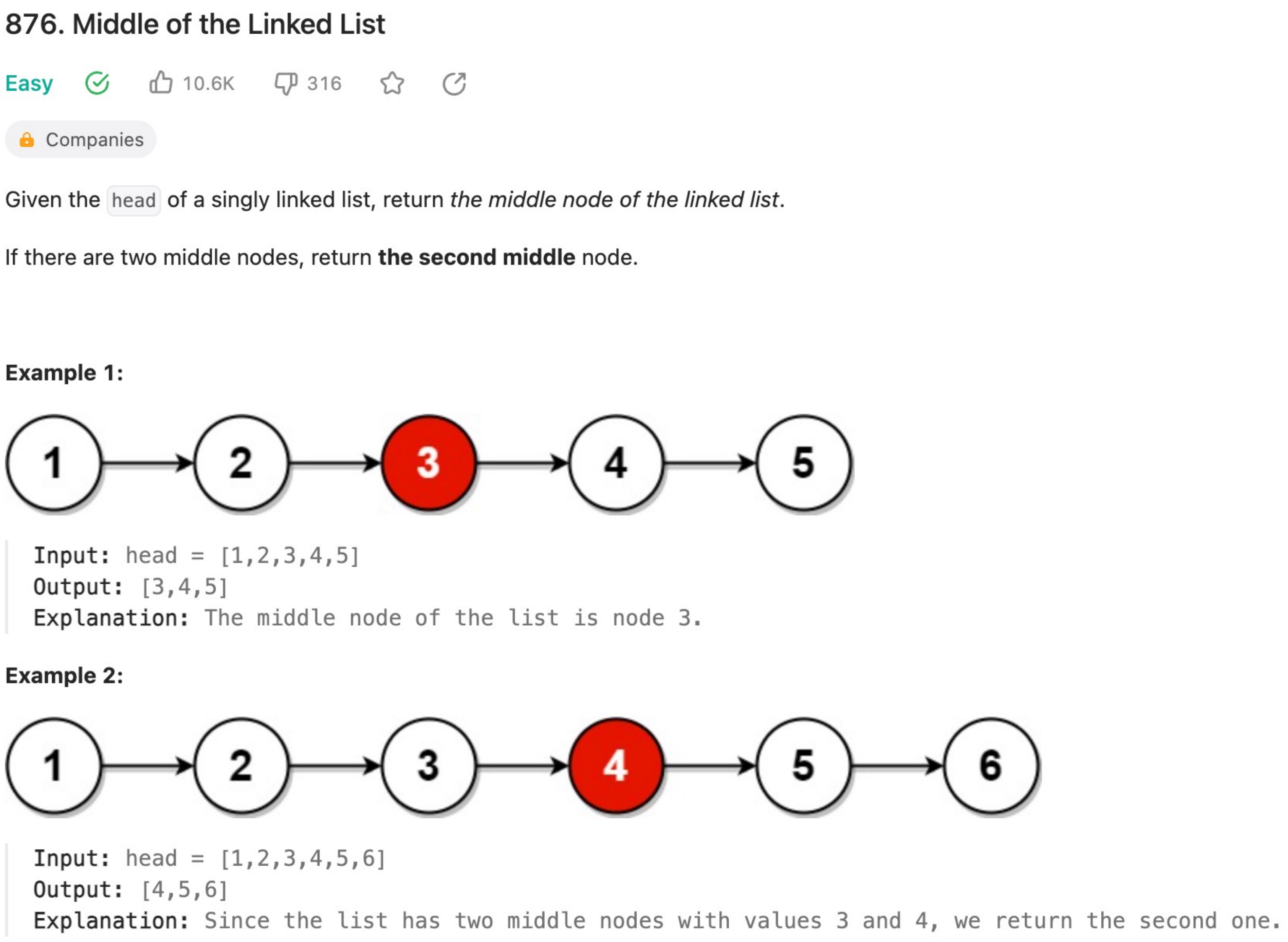


```
Input: head = [3,2,0,-4], pos = 1
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).
```

Example 2:



```
Input: head = [1,2], pos = 0
Output: true
Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.
```



876. Middle of the Linked List

Easy ✓ 👍 10.6K 👎 316 ☆

🔒 Companies

Given the `head` of a singly linked list, return *the middle node of the linked list*.

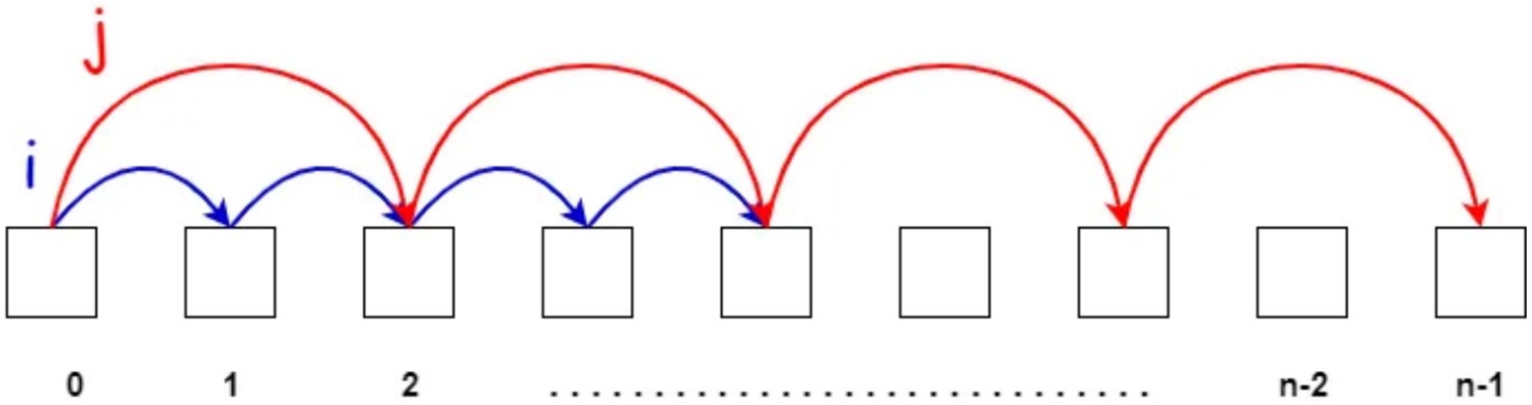
If there are two middle nodes, return **the second middle** node.

Example 1:

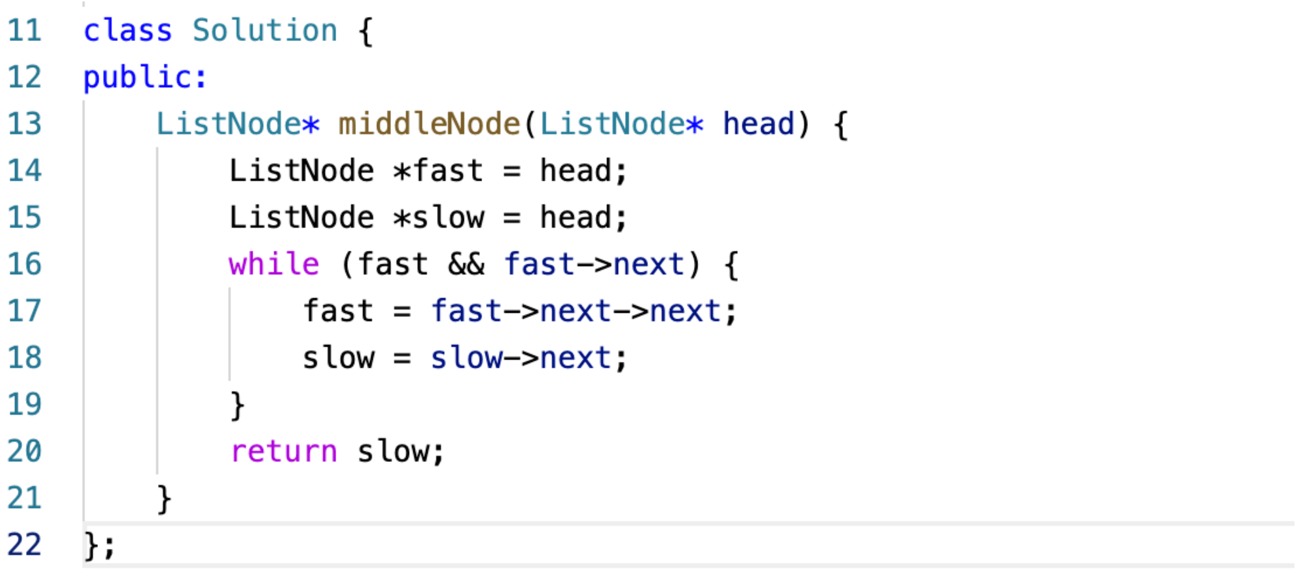
```
Input: head = [1,2,3,4,5]
Output: [3,4,5]
Explanation: The middle node of the list is node 3.
```

Example 2:

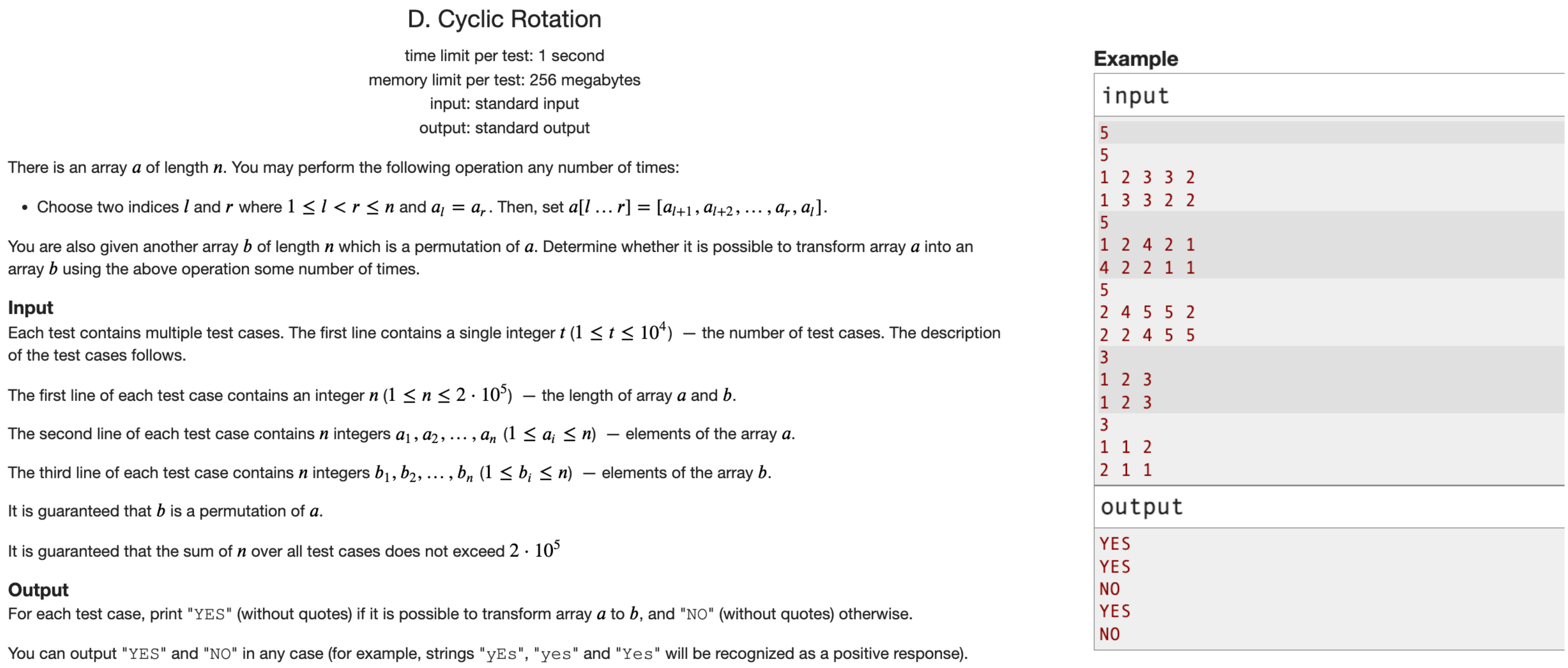
```
Input: head = [1,2,3,4,5,6]
Output: [4,5,6]
Explanation: Since the list has two middle nodes with values 3 and 4, we return the second one.
```



Source: AfterAcademy



```
class Solution {
public:
    ListNode* middleNode(ListNode* head) {
        ListNode *fast = head;
        ListNode *slow = head;
        while (fast && fast->next) {
            fast = fast->next->next;
            slow = slow->next;
        }
        return slow;
    }
};
```



D. Cyclic Rotation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is an array a of length n. You may perform the following operation any number of times:

- Choose two indices l and r where $1 \le l < r \le n$ and $a_l = a_r$. Then, set $a[l \dots r] = [a_{l+1}, a_{l+2}, \dots, a_r, a_l]$.

You are also given another array b of length n which is a permutation of a. Determine whether it is possible to transform array a into an array b using the above operation some number of times.

Input

Each test contains multiple test cases. The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n $(1 \le n \le 2 \cdot 10^5)$ — the length of array a and b.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le n)$ — elements of the array a.

The third line of each test case contains n integers $b_1, b_2, \dots, b_n$ $(1 \le b_i \le n)$ — elements of the array b.

It is guaranteed that b is a permutation of a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$

Output

For each test case, print "YES" (without quotes) if it is possible to transform array a to b, and "NO" (without quotes) otherwise.

You can output "YES" and "NO" in any case (for example, strings "yEs", "yes" and "Yes" will be recognized as a positive response).

Example

input

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

output

```
YES
YES
NO
YES
NO
```

PROBLEMAS

- https://codeforces.com/problemset/problem/1676/F
- https://codeforces.com/problemset/problem/1672/D
- https://codeforces.com/problemset/problem/1374/D
- https://codeforces.com/problemset/problem/842/A
- https://leetcode.com/problems/linked-list-cycle/
- https://www.youtube.com/watch?v=PvrxZaH_eZ4