

1686. Stone Game VI

Description

Alice and Bob take turns playing a game, with Alice starting first.

There are `n` stones in a pile. On each player's turn, they can **remove** a stone from the pile and receive points based on the stone's value. Alice and Bob may **value the stones differently**.

You are given two integer arrays of length `n`, `aliceValues` and `bobValues`. Each `aliceValues[i]` and `bobValues[i]` represents how Alice and Bob, respectively, value the `ith` stone.

The winner is the person with the most points after all the stones are chosen. If both players have the same amount of points, the game results in a draw. Both players will play **optimally**. Both players know the other's values.

Determine the result of the game, and:

- If Alice wins, return `1`.
- If Bob wins, return `-1`.
- If the game results in a draw, return `0`.

Example 1:

```
Input: aliceValues = [1,3], bobValues = [2,1]
Output: 1
Explanation:
If Alice takes stone 1 (0-indexed) first, Alice will receive 3 points.
Bob can only choose stone 0, and will only receive 2 points.
Alice wins.
```

Example 2:

```
Input: aliceValues = [1,2], bobValues = [3,1]
Output: 0
Explanation:
If Alice takes stone 0, and Bob takes stone 1, they will both have 1 point.
Draw.
```

Example 3:

```
Input: aliceValues = [2,4,3], bobValues = [1,6,7]
Output: -1
Explanation:
Regardless of how Alice plays, Bob will be able to have more points than Alice.
For example, if Alice takes stone 1, Bob can take stone 2, and Alice takes stone 0, Alice will have 6 points to Bob's 7.
Bob wins.
```

Constraints:

- `n == aliceValues.length == bobValues.length`
- `1 <= n <= 105`
- `1 <= aliceValues[i], bobValues[i] <= 100`

