Problem Statement (CircuitDesign)

A given circuit part is made of two rows of cells with a space in between them. Some of the cells of the top row need to be connected with some other cells of the bottom row with wires. A connection can be described as a pair of integers (x,y), which means that cell x of the top row needs to be connected with cell y of the bottom row.

The following picture illustrates an arrangement with the connections (1,1) (1,2) (2,2) (4,2) (4,3).

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
  |\  |    __/|
  | \ | __/ _/
  |  \|/   /
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
```

Wires have to be printed in a 2-dimensional space and cannot cross each other, so if cell x1 of the top row needs to be connected to cell y1 of the bottom row and cell x2 of the top row needs to be connected with cell y2 of the bottom row, the following must hold:

- (x1<=x2 and y1<=y2) or (x1>=x2 and y1>=y2).

Fortunately, the scientists noted that they may exchange cells placed on the same row. For instance, the following list of connections would not be valid according to the above rule: (1,2) (2,1). But, if they place the top row in the regular order [1,2] and the bottom row in reverse order [2,1], then no wires cross and the arrangement is valid. Moreover, they could also achieve that by placing the top row in reverse order and the bottom row in regular order. The following picture illustrates the situation:

```
+---+---+          +---+---+          +---+---+
| 1 | 2 |          | 1 | 2 |          | 2 | 1 |
+---+---+          +---+---+          +---+---+
  \  /               |   |              |   |
   \/     _____\     |   |              |   |
   /\          /     |   |      or      |   |
  /  \               |   |              |   |
+---+---+          +---+---+          +---+---+
| 1 | 2 |          | 2 | 1 |          | 1 | 2 |
+---+---+          +---+---+          +---+---+
```

You will be given an int **n** that represents the number of cells in each row, and the list of connections as two int[]s **top** and **bottom**. For each i, the pair (**top**[i],**bottom**[i]) is a required connection. Return the number of possible arrangements of the cells in each row that result in no wires crossing, modulo 1000000007. Two arrangements are different if any cell in any row has a different placement. See examples for further clarification.

Definition

Method signature: int countPerms(int n, int[] top, int[] bottom)

Constraints

- **n** will be between 1 and 50, inclusive.
- **top** will contain between 1 and 50 elements, inclusive.
- t**op** and **bottom** will contain the same number of elements.
- Each element of **top** and **bottom** will be between 1 and **n**, inclusive.
- Each pair (**top**[i] , **bottom**[i]) will be different.

Examples

0)

```
4

{1,1,2,4,4}

{1,2,2,2,3}

Returns: 32
```

This is the first example drawn in the problem statement.

1)

```
2

{1,2}

{2,1}

Returns: 2
```

The second example drawn in the problem statement.

2)

```
3

{1,1,1,2,2,2,3,3,3}

{1,2,3,1,2,3,1,2,3}

Returns: 0
```

There are too many wires to avoid crossing.

3)

```
30

{5,5,5,5,5,5,5,5,5, 5,11,12,13,14,15,16,
17,18,19,20,21,22,23,24,25,26,27,28,29,30}

{1,2,3,4,5,6,7,8,9,10,15,15,15,15,15,15,
Ű15,15,15,15,25,25,25,25,25,25,25,25,25,25}

Returns: 628573100
```

4)

```
5

{1,2,3,4,5,1,2,3,4}

{1,2,3,4,5,5,1,4,2}

Returns: 2
```