Problem C: Cake

Time limit: 15s, memory limit: 1GB.

Once upon a time Babushka Bajtmiła (known for the exception pierogi from problem B) baked a cake. She cut it into $2n$ rectangular pieces (two rows containing n pieces each) and put colour icing on the top of each piece. She looked at her masterpiece and was taken aback: the colour scheme looked absolutely *awful*.

Bajtmiła decided that she needs to change the colour configuration to something better. Unfortunately, changing the locations of the pieces one by one is out of question: any attempt to take out a single piece out of the cake would inevitably make its sides chipped. Babushka Bajtmiła would never serve her guests a cake which appears raggedly cut!

Fortunately, Bajtmiła possesses a rectangular spatula which can fit exactly four pieces (in two rows containing two pieces each). She can therefore carefully take four such pieces out, rotate the spatula and insert them back from the opposite side. We can formally describe this operation as taking a 2 x 2 square and rotating it by 180 degrees.

Remembering your quick and efficient help in problem B, Babushka knew exactly what to do: she asked you to determine the minimum possible number of operations that would turn her initial frosting configuration into something nice. Of course, you did what any decent person would do in your place: you said you won't be able to help because the problem is not well-defined. Bajtmiła expected that: she went to the whiteboard and draw one specific *nice* configuration and asked you to find the minimal number of operations to change her initial frosting configuration into this specific one.

It might have also happened that the granny (tired from moving all the pierogi in problem B) was mistaken and it is actually impossible to obtain the desired configuration. In such case you also need to notify her as soon as possible!

Input

The first line of input contains the number of test cases z ($1 \leqslant z \leqslant 10\,000$). The descriptions of the test cases follow.

The first line of a test case contains an integer n ($2 \leqslant n \leqslant 500\,000$). The following two lines of input describe two rows of the initial frosting configuration. Each of them contains n integers from the range $[1, 10^9]$ separated by single spaces — identifiers of frosting colours on the subsequent pieces of the cake. Note that a colour may appear more than once.

The next two lines describe the final configuration in the same format.

The total sum of n over all the test cases does not exceed $2\,000\,000$.

Output

For each test case output a single integer — the minimum possible number of operations needed to achieve the final configuration. If it is not possible to achieve the desired outcome, print -1.

Example

For an example input:	the correct output is:
2 4 1 2 3 2 4 3 1 3 3 2 1 1 2 3 4 3 2 1 2 3 4 3 4 1 2	3 -1

Explanation

In the first test case the initial configuration is

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

We can achieve the desired configuration in three steps:
1. rotating the leftmost square,

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

2. rotating the rightmost square,

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

3. rotating the middle square.

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

It is not possible to achieve the desired configuration in the second test case.