

Problem Statement (YetAnotherBingoProblem)

You might have heard of a game called Bingo. In this problem we will consider a variation of this game. Our version of Bingo is a game for two players. Each player in the game holds one to five cards. Each of the cards has 25 numbers printed on it. The numbers are integers from 1 to 75 and are pairwise distinct within each particular card.

The game consists of at most 75 turns. In each turn, a number from 1 to 75 is drawn. Numbers drawn in different turns are distinct. Both players search for the drawn number in their cards and cross out each of its occurrences. As soon as one of the players crosses out all numbers on at least one of his cards, the game ends and this player is declared the winner. However, if such a situation occurs for both players in the same turn, neither of them is declared the winner and the game ends in a draw instead.

Manao is playing Bingo against his friend. Manao sees the friend's cards and he has also hacked the number drawing machine and can make it draw the numbers in any order he needs. Surely, Manao wants to win, but he doesn't want his friend to suspect him in cheating. So he wants the numbers to be drawn in such an order that he wins and the game lasts the maximum possible number of turns.

You are given `String[]s cards1` and `cards2` representing Manao's and his friend's cards, respectively. Concatenate the elements of `cards1` to obtain a comma-separated list of space-separated lists of integers. Each of the space-separated lists represents a single card and contains the 25 numbers printed on it. `cards2` comprises the friend's cards in the same fashion. If Manao can win the game, return the maximum possible number of turns in which it can happen. Otherwise, return -1.

Definition

Class: YetAnotherBingoProblem

Method: longestWinningSequence

Parameters: `String[], String[]`

Returns: `int`

Method signature: `int longestWinningSequence(String[] cards1, String[] cards2)`

(be sure your method is public)

Constraints

- `cards1|2` will contain between 1 and 50 elements, inclusive.
- Each element of `cards1|2` will contain between 1 and 50 characters, inclusive.
- The concatenation of elements in `cards1|2` will represent a comma-separated list of at most five space-separated lists of integers. Each of these lists will contain 25 distinct numbers without leading zeros. Each of the numbers will be between 1 and 75, inclusive.

Examples

0)

```
{"60 28 ","8 25 54"," 31 13 26 71 20 66 55 72 67 18 51 52 48 63 53 ","42 16 ","1 30 56","  
"27 43 70 48 54 24 73 37 41 51 58 55"," 71 12 60 49 64 47 35 4 ","16 69 14 40 50"}  
{"5 48 26 64 12 47 39 29 63 50 15 51 33 5","8 49 73 13 61 69 16 53 41 60 ","59 55","  
"16 55 57 43 20 3 74 12 4","2 ","35 19 65 51 17 1 15 69 56"," 31 13 9 24 58"," ","38 75"}
```

Returns: 74

Both Manao and his friend hold two cards. One of the longest possible winning sequences is as follows:
13, 14, 15, ..., 75, 11, 10, 9, ..., 1.

1)

```
{"34 ","56"," 7 22 21 43 57 38 ","12 5 32"," 39 15 13"," 75 20 ","55 69 70 2","3 65 48 33 9 44"}  
  
{"46 18 3 33 34 58 24 36 30 50 ","45 11 56 25 57 60 54 74 42 8 70 40 20 2 1,26 61 20",  
" 43 52 ","2 22 33 53 44 60 27 57 12 7 15 29 73 50 ","2","8 56 34 75 67 37,47 61 36 75 45 22 17 49 33 59",  
" 48 39 69 27 3 70 12 46 54 1","1 67 ","43 73 5","2 68"}
```

Returns: 73

Manao has a single card and his friend holds three. If all the numbers from 1 to 75 are drawn in order, omitting 26 and 46, Manao wins after the 73rd turn.

2)

```
{"70 29 47 73 17 49 13 64 42 71 63 28 9 54 ","74 20 69 16 14 66 12 25 4","5 41 68"}  
{"4","7 16 68 74 71 42 49 66 45 25 13 9 12"," 17 ","20"," 5","4 70 28 64 69 73 63 14 41"," 29"}
```

Returns: -1

The set of numbers on Manao's card is the same as on his friend's card. No matter how the numbers are drawn, the game ends in a draw.

3)

```
{"2 21 15 48 50 ","75 67 33 60 52 1","4 72 64 25 8 23 39 12 11 74 65 63 56 44 13",  
"13 3","3 52 72 39 63 44 14 21 48 3","8 65 12 56"," 74 60 23 8 15 50 75",  
" 67 25 11 2,67 38 14 48 56 52 72 50 23 21 15 13 ","64 39 2 11 25 8 33 44 12 60",  
" 65 75 74,38 15 50 11 13 12 44 60 74 21 25"," 48 67 39 72 14 8 63 64 56 75 33 ","52 31 23"}  
{"7","4 21 12 60 25 38 15 11 49 ","54 8 63 33 65 56 52 13 14 72 75 23 48 50 67 3",  
"9,52 14 25 39 64 50 21 33 2 15 67 13 48 ","56 12 7","5 63 60 11 65 44 74 ",  
"23 72 29,25 12 74 2 64 31 39 33 ","14"," 21 13 72 15 23 8 60 5","6 50 38 75 ","44 46 67 1","1 48"}
```

Returns: 73