

Problem Statement (ImageTraders)

There is a community of art lovers who meet from time to time and trade images with each other. Each image transaction must obey the following rules:

1. The price at which the image is sold must be greater than or equal to the price at which it was bought.
2. No trader is allowed to buy the same image more than once.

A new image has just appeared on the market. Trader 0 bought it from an unknown artist for the price of '0', and he is now ready to trade it among his fellow art lovers. You are given a `String[] price`, where the j -th character of the i -th element is a digit representing the price at which trader j will buy the image from trader i . '0' is the lowest price, and '9' is the highest. Assuming all transactions obey the rules mentioned above, determine the longest possible sequence of transactions involving the new image. After all the transactions are done, return the number of people who have owned the image at some point in time, including both the final owner and trader 0.

Definition

Class: ImageTraders

Method: maxOwners

Parameters: `String[]`

Returns: `int`

Method signature: `int maxOwners(String[] price)`

(be sure your method is public)

Constraints

- **price** will contain exactly N elements, where N is between 2 and 15, inclusive.
- Each element of **price** will contain exactly N characters.
- Each element of **price** will contain only digits ('0'-'9').

Examples

0)

```
{
  "01",
  "10"
}
```

Returns: 2

Trader 0 can sell the image to trader 1 for the price of '1'. Both traders were owners of the image, so the answer is 2.

1)

```
{
  "022",
  "101",
  "110"
}
```

Returns: 2

We have 3 traders here. Trader 0 can sell the image to either trader 1 or trader 2 for the price of '2'. In both cases, the buyer would have to sell the image for a price of at least '2', which is impossible. Therefore, the maximal number of owners is 2.

2)

```
{  
"01231",  
"00231",  
"00031",  
"00002",  
"00000"  
}
```

Returns: 4

Here the image can have 4 owners:

- Trader 0 sells the image to trader 1 for the price of '1'.
- Trader 1 sells the image to trader 2 for the price of '2'.
- Trader 2 sells the image to trader 3 for the price of '3'.

3)

```
{  
"15555",  
"11111",  
"15111",  
"11111",  
"11111"  
}
```

Returns: 3

Trader 0 can sell the image to trader 2 for the price of '5'. Then, trader 2 can sell it to trader 1.

4)

```
{  
"0100000000",  
"0020000000",  
"0003000000",  
"0000400000",  
"0000050000",  
"0000006000",  
"0000000700",  
"0000000080",  
"0000000009",  
"1111111111"  
}
```

Returns: 10

The image can be traded among all the traders in the following sequence: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.