Amazon | OA 2020 | Five Star Sellers

823 VIEWS

Third-party companies that sell their products on [Amazon.com](http://amazon.com/) are able to analyze the customer reviews for their products in real time. Imagine that Amazon is creating a category called "five-star sellers" that will only display products sold by companies whose average percentage of five-star reviews per-product is at or above a certain threshold. Given the number of five-star and total reviews for each product a company sells, as well as the threshold percentage, what is the minimum number of additional fivestar reviews the company needs to become a five-star seller?

For example, let's say there are 3 products (n = 3) where productRatings = [[4,4], [1,2], [3, 6]], and the percentage ratings Threshold = 77. The first number for each product in productRatings denotes the number of fivestar reviews, and the second denotes the number of total reviews. Here is how we can get the seller to reach the threshold with the minimum number of additional five-star reviews:

Before we add more five-star reviews, the percentage for this seller is ((4 / 4) + (1/2) + (3/6))/3 = 66.66%  
If we add a five-star review to the second product, the percentage rises to ((4 / 4) + (2/3) +(3/6))/3 = 72.22%  
If we add another five-star review to the second product, the percentage rises to ((4 / 4) + (3/4) + (3/6))/3 = 75.00%  
If we add a five-star review to the third product, the percentage rises to ((4/4) + (3/4) + (4/7))/3 = 77.38%  
At this point, the threshold of 77% has been met. Therefore, the answer is 3 because that is the minimum number of additional five-star reviews the company needs to become a five-star seller.

Function Description

Complete the function fiveStarReviews in the editor below.

fiveStarReviews has the following parameters:

int productRatings[n][2]: a 2-dimensional array of integers where the ith element contains two values, the first one denoting fivestar[i] and the second denoting total[i]

int ratingsThreshold: the threshold percentage, which is the average percentage of five-star reviews the products need for the company to be considered a five-star seller

Returns:

int: the minimum number of additional five-star reviews the company needs to meet the threshold ratingsThreshold

Constraints

1<=n<=200  
0 <= fivestar<total<=100  
1<=ratingsThreshold<100  
The array productRatings contains only non-negative integers.

**200. Number of Islands**

Medium

Given an m x n 2d grid map of '1's (land) and '0's (water), return *the number of islands*.

An **island** is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

**Example 1:**

**Input:** grid = [

["1","1","1","1","0"],

["1","1","0","1","0"],

["1","1","0","0","0"],

["0","0","0","0","0"]

]

**Output:** 1

**Example 2:**

**Input:** grid = [

["1","1","0","0","0"],

["1","1","0","0","0"],

["0","0","1","0","0"],

["0","0","0","1","1"]

]

**Output:** 3

**Constraints:**

* m == grid.length
* n == grid[i].length
* 1 <= m, n <= 300
* grid[i][j] is '0' or '1'.

**819. Most Common Word**

Easy

Given a paragraph and a list of banned words, return the most frequent word that is not in the list of banned words.  It is guaranteed there is at least one word that isn't banned, and that the answer is unique.

Words in the list of banned words are given in lowercase, and free of punctuation.  Words in the paragraph are not case sensitive.  The answer is in lowercase.

**Example:**

**Input:**

paragraph = "Bob hit a ball, the hit BALL flew far after it was hit."

banned = ["hit"]

**Output:** "ball"

**Explanation:**

"hit" occurs 3 times, but it is a banned word.

"ball" occurs twice (and no other word does), so it is the most frequent non-banned word in the paragraph.

Note that words in the paragraph are not case sensitive,

that punctuation is ignored (even if adjacent to words, such as "ball,"),

and that "hit" isn't the answer even though it occurs more because it is banned.

**Note:**

* 1 <= paragraph.length <= 1000.
* 0 <= banned.length <= 100.
* 1 <= banned[i].length <= 10.
* The answer is unique, and written in lowercase (even if its occurrences in paragraph may have uppercase symbols, and even if it is a proper noun.)
* paragraph only consists of letters, spaces, or the punctuation symbols !?',;.
* There are no hyphens or hyphenated words.
* Words only consist of letters, never apostrophes or other punctuation symbols.

**547. Friend Circles**

There are **N** students in a class. Some of them are friends, while some are not. Their friendship is transitive in nature. For example, if A is a **direct** friend of B, and B is a **direct** friend of C, then A is an **indirect** friend of C. And we defined a friend circle is a group of students who are direct or indirect friends.

Given a **N\*N** matrix **M** representing the friend relationship between students in the class. If M[i][j] = 1, then the ith and jth students are **direct** friends with each other, otherwise not. And you have to output the total number of friend circles among all the students.

**Example 1:**

**Input:**

[[1,1,0],

[1,1,0],

[0,0,1]]

**Output:** 2

**Explanation:**The 0th and 1st students are direct friends, so they are in a friend circle.

The 2nd student himself is in a friend circle. So return 2.

**Example 2:**

**Input:**

[[1,1,0],

[1,1,1],

[0,1,1]]

**Output:** 1

**Explanation:**The 0th and 1st students are direct friends, the 1st and 2nd students are direct friends,

so the 0th and 2nd students are indirect friends. All of them are in the same friend circle, so return 1.

**Constraints:`**

* 1 <= N <= 200
* M[i][i] == 1
* M[i][j] == M[j][i]