LeetCode\_599\_MinimumIndexSumOfTwoLists

# LeetCode\_599\_MinimumIndexSumOfTwoLists

## 题目介绍

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LeetCode\_599\_MinimumIndexSumOfTwoLists\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* 难度：Easy

\* DateTime：2018-10-06 10:50

\* https://leetcode.com/problemset/all/?topicSlugs=hash-table

\* 题目介绍：

\* Suppose Andy and Doris want to choose a restaurant for dinner,

\* and they both have a list of favorite restaurants represented by strings.

\*

\* You need to help them find out their common interest

\* with the least list index sum. If there is a choice tie between answers,

\* output all of them with no order requirement. You could assume there always exists an answer.

\* Example 1:

\* Input:

\* ["Shogun", "Tapioca Express", "Burger King", "KFC"]

\* ["Piatti", "The Grill at Torrey Pines", "Hungry Hunter Steakhouse", "Shogun"]

\* Output: ["Shogun"]

\* Explanation: The only restaurant they both like is "Shogun".

\* Example 2:

\* Input:

\* ["Shogun", "Tapioca Express", "Burger King", "KFC"]

\* ["KFC", "Shogun", "Burger King"]

\* Output: ["Shogun"]

\* Explanation: The restaurant they both like and have the least index sum is "Shogun" with index sum 1 (0+1).

\* Note:

\* The length of both lists will be in the range of [1, 1000].

\* The length of strings in both lists will be in the range of [1, 30].

\* The index is starting from 0 to the list length minus 1.

\* No duplicates in both lists.

## 思路分析

\* 思路分析：结果一定是一个吗？不一定，如['A','B'] 和['B','A']结果就是两个。

\* 首先将A第一个人喜欢的餐厅放入一个HashMap中，key就是餐厅String，value就是

\* 对应的下标；然后逐个判断第二个人喜欢的餐厅，然后将下标值加在一起；

\* 下标和初始化为Integer.MAX\_VALUE；

\* 若新的下标和比之前的大，直接continue；

\* 若新的下标和与之前的相等，则将该结果添加至结果集；

\* 若新的下标和比之前的小，结果集先清空，再将该餐厅加入结果集，记得要更新minIndexSum。

## Java代码

/\*\*

\* 基于HashMap解决

\*/

public String[] findRestaurant(String[] list1, String[] list2) {

if(list1.length == 0||list2.length == 0) return new String[]{};

HashMap<String,Integer> hashMap = new HashMap<String,Integer>() ;

for(int i = 0;i < list1.length;i++ )

hashMap.put(list1[i],i);

List<String> res = new ArrayList<String>();

int minIndexSum = Integer.MAX\_VALUE;

for(int i = 0;i < list2.length;i++){

Integer index1 = hashMap.get(list2[i]);

if(index1 == null || index1 + i > minIndexSum) continue;

if(index1 + i == minIndexSum) res.add(list2[i]);

else{//index1 + i < indexSum

res.clear();//清空

res.add(list2[i]);

minIndexSum = index1 + i;//这里需要更新indexSum

}

}

return res.toArray(new String[res.size()]);//List==>数组

}