Minimum Index Sum of Two Lists

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" w
ith index sum 1 (0+1).
```

Note:

1. The length of both lists will be in the range of [1, 1000].
2. The length of strings in both lists will be in the range of [1, 30].
3. The index is starting from 0 to the list length minus 1.
4. No duplicates in both lists.

Solution 1

```
public String[] findRestaurant(String[] list1, String[] list2) {
    Map<String, Integer> map = new HashMap<>();
    List<String> res = new LinkedList<>();
    int minSum = Integer.MAX_VALUE;
    for (int i=0;i<list1.length;i++) map.put(list1[i], i);
    for (int i=0;i<list2.length;i++) {
        Integer j = map.get(list2[i]);
        if (j != null && i + j <= minSum) {
            if (i + j < minSum) { res = new LinkedList<>(); minSum = i+j; }
            res.add(list2[i]);
        }
    }
    return res.toArray(new String[res.size()]);
}
```

written by compton_scatter original link here

Solution 2

Say the lists are `A` and `B`. Let `Aindex[element]` be the index of that element in A. For every index, value pair (j, v) in B, we have some candidate sum-of-indexes i + j, where i = Aindex[v] if it exists. If the candidate sum is better, it becomes our new answer; if the candidate sums are the same, then we append to our answer.

```
def findRestaurant(self, A, B):
    Aindex = {u: i for i, u in enumerate(A)}
    best, ans = 1e9, []

    for j, v in enumerate(B):
        i = Aindex.get(v, 1e9)
        if i + j < best:
            best = i + j
            ans = [v]
        elif i + j == best:
            ans.append(v)
    return ans
```

written by awice original link here

Solution 3

```java
public class Solution {
    public String[] findRestaurant(String[] list1, String[] list2) {
        List<String> result = null;
        Map<String, Integer> map1 = new HashMap<>();
        Map<String, Integer> map2 = new HashMap<>();
        int min = Integer.MAX_VALUE;

        for (int i = 0; i < list1.length; i++) {
            map1.put(list1[i], i);
        }
        for (int i = 0; i < list2.length; i++) {
            map2.put(list2[i], i);
        }

        for (int i = 0; i < list1.length; i++) {
            if (map2.containsKey(list1[i])) {
                int sum = map1.get(list1[i]) + map2.get(list1[i]);
                if (sum < min) {
                    min = sum;
                    result = new ArrayList<String>();
                    result.add(list1[i]);
                }
                else if (sum == min) {
                    result.add(list1[i]);
                }
            }
        }

        String[] res = new String[result.size()];
        for (int i = 0; i < result.size(); i++) {
            res[i] = result.get(i);
        }

        return res;
    }
}
```

written by shawngao original link here

From Leetcoder.