👍 205   👎 97   ♡   ▼

760. Find Anagram Mappings

📋 Notes

🗒 Description (/problems/find-anagram-mappings/description/)	💡 Hints (/problems/find-anagram-mappings/hints/)	Submissions (/proble

🔀 Pick One (/problems/random-one-question/)

Given two lists `A` and `B`, and `B` is an anagram of `A`. `B` is an anagram of `A` means `B` is made by randomizing the order of the elements in `A`.

We want to find an *index mapping* `P`, from `A` to `B`. A mapping `P[i] = j` means the `i` th element in `A` appears in `B` at index `j`.

These lists `A` and `B` may contain duplicates. If there are multiple answers, output any of them.

For example, given

```
A = [12, 28, 46, 32, 50]
B = [50, 12, 32, 46, 28]
```

We should return

```
[1, 4, 3, 2, 0]
```

as `P[0] = 1` because the `0` th element of `A` appears at `B[1]`, and `P[1] = 4` because the `1` st element of `A` appears at `B[4]`, and so on.

Note:

1. `A, B` have equal lengths in range `[1, 100]`.
2. `A[i], B[i]` are integers in range `[0, 10^5]`.

Seen this question in a real interview before? Yes No   ✖

Companies ℹ ▲   🕐

Google 2 (/company/google)

Related Topics ▲

Hash Table (/tag/hash-table)

| Java   ▼ | | ⟨⟩ | ⟳ | >_ | ⚙ |

```java
1 ▾ class Solution {
2 ▾     public int[] anagramMappings(int[] A, int[] B) {
3           Map<Integer, Deque<Integer>> valueToIndices = new HashMap<>();
```

```
 4 ▾        for (int i=0; i< B.length; i++) {
 5 ▾            Integer value = B[i];
 6 ▾            if (!valueToIndices.containsKey(value)) {
 7                    valueToIndices.put(value, new LinkedList<>());
 8                }
 9                valueToIndices.get(value).addLast(i);
10            }
11
12 ▾        int[] mapping = new int[A.length];
13 ▾        for (int i=0; i<A.length; i++) {
14 ▾          mapping[i] = valueToIndices.get(A[i]).pollFirst();
15            }
16            return mapping;
17        }
18  }
```

☐ **Custom Testcase** (**Contribute ❶**)

▶ Run   ⬆ Submit

Submission Result: Accepted (/submissions/detail/171930852/) ❓

More Details ❯ (/submissions/detail/171930852/)

Next challenges:

H-Index (/problems/h-index)   Sentence Similarity (/problems/sentence-similarity)   Maximum Frequency Stack (/problems/maximum-frequency-stack)

Share your acceptance!

◀ 7   ◀ 5

Check out our solution!   Reveal Solution ❯ (/articles/find-anagram-mappings/)

Copyright © 2018 LeetCode

Contact Us (/support/) | Frequently Asked Questions (/faq/) | Terms of Service (/terms/) | Privacy Policy (/privacy/)

🇺🇸 United States (/region/)