813. Find Anagram Mappings

* [Description](http://lintcode.com/en/problem/find-anagram-mappings/" \l "description)
* [Notes](http://lintcode.com/en/problem/find-anagram-mappings/#note)
* [Testcase](http://lintcode.com/en/problem/find-anagram-mappings/#testcase)
* [Judge](http://lintcode.com/en/problem/find-anagram-mappings/#judge)

Given two lists A and B, and B is an anagram of A. B is an anagram of A means Bis 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 ith element in Aappears in B at index j.

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

https://leetcode.com/problems/find-anagram-mappings/description/

 Notice

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

Have you met this question in a real interview?

Yes

**Example**

Given A = [12, 28, 46, 32, 50] and B = [50, 12, 32, 46, 28], return [1, 4, 3, 2, 0].

Explanation:

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

<http://lintcode.com/en/problem/find-anagram-mappings/#>

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication1;

import java.util.\*;

/\*\*

\*

\* @author Usuario

\*/

public class JavaApplication1 {

public static int[] anagramMappings(int[] A, int[] B) {

// Write your code here

int[] res = new int[A.length];

HashMap<Integer, Integer> bMap = new HashMap();

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

bMap.put(B[i], i);

}

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

res[i] = bMap.get(A[i]);

}

return res;

}

public static void main(String[] args) {

// TODO code application logic here

int[] A = {12, 28, 46, 32, 50};

int[] B = {50, 12, 32, 46, 28};

int[] res = anagramMappings(A, B);

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

System.out.print(res[i] + " ");

}

}

}