**[Find Anagram Mappings](https://leetcode.com/problems/find-anagram-mappings/)**

**Brute Force Solution:**

**public** **class** MappingAnagram {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

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

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

**int** [] result = *anagramMappings*(A, B);

**for**(**int** i : result)

System.***out***.print(i + " ");

}

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

**int** [] result = **new** **int**[A.length];

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

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

**if**(A[i] == B[j]) {

result[i] = j;

**break**;

}

}

}

**return** result;

}

}

Time Complexity : O(n^2) , n is number of elements in array.

Space Complexity : O(n), n is number of elements in array.

**HashMap Solution(Efficient):**

**import** java.util.HashMap;

**public** **class** MappingAnagram {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

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

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

result = *anagramMappingsHashMap*(A, B);

System.***out***.println(" ");

**for**(**int** i : result)

System.***out***.print(i + " ");

}

**public** **static** **int**[] anagramMappingsHashMap(**int**[] A, **int**[] B) {

**int** [] result = **new** **int**[A.length];

HashMap<Integer, Integer> mapping = **new** HashMap<>();

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

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

}

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

**if**(mapping.containsKey(A[i])) {

result[j++] = mapping.get(A[i]);

}

}

**return** result;

}

}

Time Complexity : O(2n) , n is number of elements in array.

Space Complexity : O(2n), n is number of elements in array.