2215. Find the Difference of Two Arrays

Problem Statement

Check the problem statement here.

Java Solution

My Solution

```java
class Solution {
    public List<List<Integer>> findDifference(int[] nums1, int[] nums2) {
        List<List<Integer>> ans = new ArrayList<List<Integer>>();
        List<Integer> ans1 = new ArrayList<>();
        List<Integer> ans2 = new ArrayList<>();

        for(int i=0; i<nums1.length; i++) {
            int flag = 0;
            if(nums1[i]>1000){
                continue;
            }
            for(int j=0; j<nums2.length; j++) {
                if(nums1[i]==nums2[j]){
                    flag+=1;
                    nums2[j] = 1001;
                }
            }

            if(flag==0){
                // check if it is duplicated
                int flag1 = 0;
                for(int ii: ans1){
                    if(ii==nums1[i]){
                        flag1+=1;
                        break;
                    }
                }
                if(flag1==0){
                    ans1.add(nums1[i]);
                }
            }

            int temp = nums1[i];
            for(int iii=i; iii<nums1.length; iii++) {
                if(temp==nums1[iii]){
                    nums1[iii]=1001;
                }
            }
        }

        for(int i=0; i<nums2.length; i++) {
            if (nums2[i]<=1000){
                int flag2 = 0;
                for(int ii: ans2){
                    if(ii==nums2[i]){
                        flag2+=1;
                        break;
                    }
                }
                if(flag2==0){
                    ans2.add(nums2[i]);
                }
            }
        }
        ans.add(0,ans1);
        ans.add(1,ans2);

        return ans;
    }
}
```

Time Complexity

O(n*m) where n is the length of nums1 and m is the length of nums2.

Space Complexity

O(1) except the space for the answer.

Using HashMap

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class Solution {
    public List<List<Integer>> findDifference(int[] nums1, int[] nums2) {
        HashMap<Integer, Integer> map = new HashMap<>();
        List<Integer> list1 = new ArrayList<>();
        List<Integer> list2 = new ArrayList<>();

        // Add all elements of nums1 to the map
        for (int num1 : nums1) {
            map.put(num1, map.getOrDefault(num1, 0) + 1);
        }

        // Check each element in nums2 for uniqueness against nums1
        for (int num2 : nums2) {
            if (!map.containsKey(num2) && !list2.contains(num2)) {
                list2.add(num2);
            }
        }
        // Check each element in nums1 for uniqueness against nums2
        for (int num1 : nums1) {
            if (map.get(num1) == 1 && !list1.contains(num1)) {
                list1.add(num1);
            }
        }

        List<List<Integer>> result = new ArrayList<>();
        result.add(list1);
        result.add(list2);
        return result;
    }

    public static void main(String[] args) {
        Solution solution = new Solution();
        int[] nums1 = {4, 9, 5};
        int[] nums2 = {9, 4, 9, 8, 4};
        List<List<Integer>> result = solution.findDifference(nums1,
nums2);
        System.out.println(result);

    }
}
```

Time Complexity

O(n+m) where n is the length of nums1 and m is the length of nums2.

Space Complexity

O(n+m) where n is the length of nums1 and m is the length of nums2.

Using HashSet

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;

public class Solution {
    public List<List<Integer>> findDifference(int[] nums1, int[] nums2) {
        HashSet<Integer> set1 = new HashSet<>();
        HashSet<Integer> set2 = new HashSet<>();

        // Add all elements of nums1 to the set
        for (int num1 : nums1) {
            set1.add(num1);
        }

        // Check each element in nums2 for uniqueness against nums1
        for (int num2 : nums2) {
            if (!set1.contains(num2) && !set2.contains(num2)) {
                set2.add(num2);
            }
        }

        // Check each element in nums1 for uniqueness against nums2
        for (int num1 : nums1) {
            if (!set2.contains(num1)) {
                set1.remove(num1);
            }
        }

        List<List<Integer>> result = new ArrayList<>();
        result.add(new ArrayList<>(set1));
        result.add(new ArrayList<>(set2));
        return result;
    }

    public static void main(String[] args) {
        Solution solution = new Solution();
        int[] nums1 = {4, 9, 5};
        int[] nums2 = {9, 4, 9, 8, 4};
        List<List<Integer>> result = solution.findDifference(nums1,
nums2);
        System.out.println(result);

    }
}
```

Time Complexity

O(n+m) where n is the length of nums1 and m is the length of nums2.

Space Complexity

O(n+m) where n is the length of nums1 and m is the length of nums2.

Using ArraryList Without HashMap

```
import java.util.ArrayList;
import java.util.List;

public class Solution {
    public List<List<Integer>> findDifference(int[] nums1, int[] nums2) {
        List<Integer> list1 = new ArrayList<>();
        List<Integer> list2 = new ArrayList<>();

        // Check each element in nums1 for uniqueness against nums2
        for (int num1 : nums1) {
            if (!contains(nums2, num1) && !list1.contains(num1)) {
                list1.add(num1);
            }
        }

        // Check each element in nums2 for uniqueness against nums1
        for (int num2 : nums2) {
            if (!contains(nums1, num2) && !list2.contains(num2)) {
                list2.add(num2);
            }
        }

        List<List<Integer>> result = new ArrayList<>();
        result.add(list1);
        result.add(list2);
        return result;
    }

    // Helper method to check if an array contains a specific element
    private boolean contains(int[] array, int key) {
        for (int element : array) {
            if (element == key) {
                return true;
            }
        }
        return false;
    }

    public static void main(String[] args) {
        Solution solution = new Solution();
        int[] nums1 = {4, 9, 5};
        int[] nums2 = {9, 4, 9, 8, 4};
        List<List<Integer>> result = solution.findDifference(nums1,
nums2);
        System.out.println(result);

    }
}
```

Python Solution

Basic Solution

```
class Solution:
    def findDifference(self, nums1: List[int], nums2: List[int]) ->
List[List[int]]:
        ans1 = []
        ans2 = []
        for i in nums1:
            if i not in nums2 and i not in ans1:
                ans1.append(i)
        for i in nums2:
            if i not in nums1 and i not in ans2:
                ans2.append(i)
        return [ans1, ans2]
```

Time Complexity

O(n*m) where n is the length of nums1 and m is the length of nums2.

Space Complexity

O(1) except the space for the answer.

Using Set

```
class Solution:
    def findDifference(self, nums1: List[int], nums2: List[int]) -> 
List[List[int]]:
        set1 = set(nums1)
        set2 = set(nums2)
        ans1 = list(set1 - set2)
        ans2 = list(set2 - set1)
        return [ans1, ans2]
```

Time Complexity

O(n+m) where n is the length of nums1 and m is the length of nums2.

Space Complexity

o(n+m) where n is the length of nums1 and m is the length of nums2.