349. Intersection of Two Arrays (/problems/intersection-of-two-arrays/)

March 1, 2019 | 28.3K views

Average Rating: 4.29 (28 votes)

Given two arrays, write a function to compute their intersection.

Example 1:

```
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2]
```

Example 2:

```
Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [9,4]
```

Note:

- Each element in the result must be unique.
- The result can be in any order.

Solution

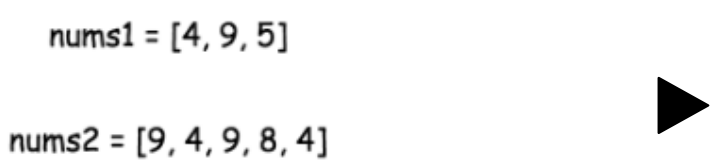
Approach 1: Two Sets

Intuition

The naive approach would be to iterate along the first array `nums1` and to check for each value if this value in `nums2` or not. If yes - add the value to output. Such an approach would result in a pretty bad $\mathcal{O}(n \times m)$ time complexity, where `n` and `m` are arrays' lengths.

> To solve the problem in linear time, let's use the structure `set`, which provides `in/contains` operation in $\mathcal{O}(1)$ time in average case.

The idea is to convert both arrays into sets, and then iterate over the smallest set checking the presence of each element in the larger set. Time complexity of this approach is $\mathcal{O}(n + m)$ in the average case.



1 / 6

Implementation

Java Python Copy

```java
class Solution {
  public int[] set_intersection(HashSet<Integer> set1, HashSet<Integer> set2) {
    int [] output = new int[set1.size()];
    int idx = 0;
    for (Integer s : set1)
      if (set2.contains(s)) output[idx++] = s;

    return Arrays.copyOf(output, idx);
  }

  public int[] intersection(int[] nums1, int[] nums2) {
    HashSet<Integer> set1 = new HashSet<Integer>();
    for (Integer n : nums1) set1.add(n);
    HashSet<Integer> set2 = new HashSet<Integer>();
    for (Integer n : nums2) set2.add(n);

    if (set1.size() < set2.size()) return set_intersection(set1, set2);
    else return set_intersection(set2, set1);
  }
}
```

Complexity Analysis

- Time complexity : $\mathcal{O}(n+m)$, where `n` and `m` are arrays' lengths. $\mathcal{O}(n)$ time is used to convert `nums1` into set, $\mathcal{O}(m)$ time is used to convert `nums2`, and `contains/in` operations are $\mathcal{O}(1)$ in the average case.
- Space complexity : $\mathcal{O}(m+n)$ in the worst case when all elements in the arrays are different.

Approach 2: Built-in Set Intersection

Intuition

There are built-in intersection facilities, which provide $\mathcal{O}(n+m)$ time complexity in the average case and $\mathcal{O}(n \times m)$ time complexity in the worst case.

> In Python it's intersection operator (https://wiki.python.org/moin/TimeComplexity#set), in Java - retainAll() function (https://docs.oracle.com/javase/8/docs/api/java/util/AbstractCollection.html#retainAll-java.util.Collection-).

Implementation

Java Python Copy

```java
class Solution {
  public int[] intersection(int[] nums1, int[] nums2) {
    HashSet<Integer> set1 = new HashSet<Integer>();
    for (Integer n : nums1) set1.add(n);
    HashSet<Integer> set2 = new HashSet<Integer>();
    for (Integer n : nums2) set2.add(n);

    set1.retainAll(set2);

    int [] output = new int[set1.size()];
    int idx = 0;
    for (int s : set1) output[idx++] = s;
    return output;
  }
}
```

Complexity Analysis

- Time complexity : $\mathcal{O}(n+m)$ in the average case and $\mathcal{O}(n \times m)$ in the worst case when load factor is high enough (https://wiki.python.org/moin/TimeComplexity#set).
- Space complexity : $\mathcal{O}(n+m)$ in the worst case when all elements in the arrays are different.

Analysis written by @liaison (https://leetcode.com/liaison/) and @andvary (https://leetcode.com/andvary/)

Rate this article:

Previous (/articles/count-univalue-subtrees/) Next (/articles/sliding-window-maximum/)

Comments: 12

Sort By

Type comment here... (Markdown is supported)

Preview Post

hasanseirafi (hasanseirafi) 93 March 9, 2019 3:10 AM

This is a Facebook interview question.
They ask for the intersection, which has a trivial solution using a hash or a set.

Then they ask you to solve it under these constraints:
O(n) time and O(1) space (the resulting array of intersections is not taken into consideration).

Read More

90 Share Reply

SHOW 9 REPLIES

JAMESJJ78 (jamesjj78) ★ 58 April 10, 2019 8:15 PM

Sometimes I feel like a rocket scientist when I see all the maths formulas

25 Share Reply

dmironov (dmironov) ★ 12 March 7, 2019 7:16 PM

Python 3 (straightforward)

```
return list(set(nums2)-(set(nums2) - set(nums1)))
```

7 Share Reply

sofs1 (sofs1) ★ 257 March 28, 2019 2:30 AM

Man, this sounds like a simple problem. But, easy to make lot of silly mistakes.

4 Share Reply

vaishnav6887 (vaishnav6887) ★ 3 June 13, 2019 11:17 AM

You may not need to store the data in Set for both the arrays. I stored element of nums1 (array with less number of elements) into Set and then iterate nums2. If element found, remove the element from the set, and add the current element to an Array to be returned.

Here is my javascript solution.

Read More

3 Share Reply

dragonpolice (dragonpolice) ★ 44 April 29, 2019 2:04 PM

Why in the end of the first approach, it returns Arrays.copyOf(output, idx) instead of output array directly?

2 Share Reply

SHOW 1 REPLY

chenzhuo2507070974 (chenzhuo2507070974) ★ 2 March 9, 2019 3:03 PM

cool~

1 Share Reply

steveo (steveo) ★ 21 March 6, 2019 3:35 PM

for solution 1, there's no need to compare lenth

0 Share Reply

SHOW 3 REPLIES

DmytroLy (dmytroly) ★ 0 May 23, 2019 6:59 AM

js

var intersection = function(nums1, nums2) {
const intersected = [];

Read More

0 Share Reply

NideeshT (nideesht) 115 April 30, 2019 6:22 AM

Java Code + Youtube Video Explanation - accepted https://youtu.be/OBZt66L4WD8 (https://youtu.be/OBZt66L4WD8) (clickable link)

Read More

0 Share Reply

1 2

Help Center (/support/) | Terms (/terms/) | Privacy Policy (/privacy/)

United States (/region/)