链滴

[每日 LeetCode] 350. Intersection of Two Arrays II

作者：Hanseltu

原文链接：https://ld246.com/article/1560959950718

来源网站：链滴

原文链接 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.tuhaoxin.c
%2Farticles%2F2019%2F06%2F19%2F1560959895117.html" target="_blank" rel="nofollow ug
">[每日 LeetCode] 350. Intersection of Two Arrays II
Description:
Given two arrays, write a function to compute their intersection.
Example 1:
<pre><code class="highlight-chroma"><span class="highlight
cl">Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2,2]
</code></pre>
Example 2:
<pre><code class="highlight-chroma"><span class="highlight
cl">Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [4,9]
</code></pre>
Note:

Each element in the result should appear as many times as it shows in both arrays.
The result can be in any order.

Follow up:

What if the given array is already sorted? How would you optimize your algorithm?
What if_nums1_’ s size is small compared to_nums2_’ s size? Which algorithm is better?
/li>
What if elements of_nums2_are stored on disk, and the memory is limited such that you c
nnot load all elements into the memory at once?

<hr>
思路：本题是<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.tuhaoxin.
n%2Farticles%2F2019%2F06%2F19%2F1560958327722.html" target="_blank" rel="nofollow
gc">[每日 LeetCode] 349. Intersection of Two Arrays 的改进版，本题要求相同的元素全部
回，而 349 题只要求返回重复中的一个。有两种思路，如下：

思路一：使用 unordered_map 容器，对 num1 中的元素为 key 和出现的次数为 value 进行统
，然后遍历 num2，在 map 中查找，如找到则对应元素的 value 减一，直到遍历完成。

思路二：可以首先对两个数组进行排序，然后使用两个指针 i,j 同时从最小的数开始，如果相等
加入到 res 数组中，若指向 num1 的元素更大，则让 j+1，否则 i+1，直到两个指针到达最大值位置

<hr>
C++ 代码（思路一）
<pre><code class="highlight-chroma"><span class="highlight
cl">class Solution {
public:
    vector<int&g
; intersect(vector<int>& nums1, vector<int>& nums2) {
        unordered_
ap<int, int> m;
        vector<int

```
gt; res;
</span></span><span class="highlight-line"><span class="highlight-cl">        for (auto a :
ums1)
</span></span><span class="highlight-line"><span class="highlight-cl">        ++m[a];
</span></span><span class="highlight-line"><span class="highlight-cl">        for (auto a :
ums2) {
</span></span><span class="highlight-line"><span class="highlight-cl">            if (m[a]--
gt; 0)
</span></span><span class="highlight-line"><span class="highlight-cl">        res.push_bac
(a);
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">        return res;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span></code></pre>
<p>运行时间：8ms</p>
<p>运行内存：9.7M</p>
<hr>
<p>C++ 代码（思路二）</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">class Solution {
</span></span><span class="highlight-line"><span class="highlight-cl">public:
</span></span><span class="highlight-line"><span class="highlight-cl">    vector&lt;int&g
; intersect(vector&lt;int&gt;&amp; nums1, vector&lt;int&gt;&amp; nums2) {
</span></span><span class="highlight-line"><span class="highlight-cl">        vector&lt;int
gt; res;
</span></span><span class="highlight-line"><span class="highlight-cl">        int i = 0, j = 0

</span></span><span class="highlight-line"><span class="highlight-cl">        sort(nums1.b
gin(), nums1.end());
</span></span><span class="highlight-line"><span class="highlight-cl">        sort(nums2.b
gin(), nums2.end());
</span></span><span class="highlight-line"><span class="highlight-cl">        while (i &lt;
ums1.size() &amp;&amp; j &lt; nums2.size()) {
</span></span><span class="highlight-line"><span class="highlight-cl">            if (nums1[i]
== nums2[j]) {
</span></span><span class="highlight-line"><span class="highlight-cl">                res.push
back(nums1[i]);
</span></span><span class="highlight-line"><span class="highlight-cl">                ++i; ++j

</span></span><span class="highlight-line"><span class="highlight-cl">            } else if (n
ms1[i] &gt; nums2[j]) {
</span></span><span class="highlight-line"><span class="highlight-cl">                ++j;
</span></span><span class="highlight-line"><span class="highlight-cl">            } else {
</span></span><span class="highlight-line"><span class="highlight-cl">                ++i;
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">        return res;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span></code></pre>
<p>运行时间：8ms</p>
<p>运行内存：8.8M</p>
<p><a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.tuhaoxin.cn%2Ftags%
```

FLeetCode" target="_blank" rel="nofollow ugc">LeetCodeSortEasy
旧一篇
<h2 id="评论">评论</h2>
<h2 id="发表评论">发表评论</h2>
<h4 id="相关阅读">相关阅读</h4>

[每日 LeetCode] 349. Intersection of Two Array
[每日 LeetCode] 543. Diameter of Binary Tree
[每日 LeetCode] 559. Maximum Depth of Nary Tree
[每日 LeetCode] 101. Symmetric Tree

<h4 id="随机阅读-">随机阅读：</h4>

[每日 LeetCode] 532. K-diff Pairs in an Array</l>
[每日 LeetCode] 665. Non-decreasing Array

[每日 LeetCode] 840. Magic Squares In Grid</l>
[每日 LeetCode] 35. Search Insert Position
[每日 LeetCode] 234. Palindrome Linked List</l>

Description:
Given two arrays, write a function to compute their intersection.
Example 1:
<pre><code class="highlight-chroma">Input: nums1 = [1,2,2,1], nums2 = [2,2]

Output: [2,2]
</code></pre>
Example 2:
<pre><code class="highlight-chroma"><span class="highlight
cl">Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [4,9]
</code></pre>
Note:

Each element in the result should appear as many times as it shows in both arrays.
The result can be in any order.

Follow up:

What if the given array is already sorted? How would you optimize your algorithm?
What if nums1 's size is small compared to nums2 's size? Whi
h algorithm is better?
What if elements of nums2 are stored on disk, and the memory is limited su
h that you cannot load all elements into the memory at once?

<hr>
思路：本题是<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.tuhaoxin.
n%2Farticles%2F2019%2F06%2F19%2F1560958327722.html" target="_blank" rel="nofollow
gc">[每日 LeetCode] 349. Intersection of Two Arrays 的改进版，本题要求相同的元素全部
回，而 349 题只要求返回重复中的一个。有两种思路，如下：

思路一：使用 unordered_map 容器，对 num1 中的元素为 key 和出现的次数为 value 进行统
，然后遍历 num2，在 map 中查找，如找到则对应元素的 value 减一，直到遍历完成。

思路二：可以首先对两个数组进行排序，然后使用两个指针 i,j 同时从最小的数开始，如果相等
加入到 res 数组中，若指向 num1 的元素更大，则让 j+1，否则 i+1，直到两个指针到达最大值位置

<hr>
C++ 代码（思路一）
<pre><code class="highlight-chroma"><span class="highlight
cl">class Solution {
public:
    vector<int&g
; intersect(vector<int>& nums1, vector<int>& nums2) {
        unordered_
ap<int, int> m;
        vector<int
gt; res;
        for (auto a :
ums1)
            ++m[a];
        for (auto a :
ums2) {
            if (m[a]--
gt; 0)

```
</span></span><span class="highlight-line"><span class="highlight-cl">        res.push_bac
(a);
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">        return res;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span></code></pre>
<p>运行时间：8ms</p>
<p>运行内存：9.7M</p>
<hr>
<p>C++ 代码（思路二）</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">class Solution {
</span></span><span class="highlight-line"><span class="highlight-cl">public:
</span></span><span class="highlight-line"><span class="highlight-cl">    vector&lt;int&g
; intersect(vector&lt;int&gt;&amp;amp; nums1, vector&lt;int&gt;&amp;amp; nums2) {
</span></span><span class="highlight-line"><span class="highlight-cl">        vector&lt;int
gt; res;
</span></span><span class="highlight-line"><span class="highlight-cl">        int i = 0, j = 0

</span></span><span class="highlight-line"><span class="highlight-cl">        sort(nums1.b
gin(), nums1.end());
</span></span><span class="highlight-line"><span class="highlight-cl">        sort(nums2.b
gin(), nums2.end());
</span></span><span class="highlight-line"><span class="highlight-cl">        while (i &lt;
ums1.size() &amp;amp;&amp;amp; j &lt; nums2.size()) {
</span></span><span class="highlight-line"><span class="highlight-cl">            if (nums1[i]
== nums2[j]) {
</span></span><span class="highlight-line"><span class="highlight-cl">                res.push
back(nums1[i]);
</span></span><span class="highlight-line"><span class="highlight-cl">                ++i; ++j

</span></span><span class="highlight-line"><span class="highlight-cl">            } else if (n
ms1[i] &gt; nums2[j]) {
</span></span><span class="highlight-line"><span class="highlight-cl">                ++j;
</span></span><span class="highlight-line"><span class="highlight-cl">            } else {
</span></span><span class="highlight-line"><span class="highlight-cl">                ++i;
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">        return res;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span></code></pre>
<p>运行时间：8ms</p>
<p>运行内存：8.8M</p>
```