

```

1  /*
2  给你两个数组，arr1 和 arr2，
3      arr2 中的元素各不相同
4      arr2 中的每个元素都出现在 arr1 中
5  对 arr1 中的元素进行排序，使 arr1 中项的相对顺序和 arr2 中的相对顺序相同。未在 arr2 中
   出现过的元素需要按照升序放在 arr1 的末尾。
6  示例：
7  输入：arr1 = [2,3,1,3,2,4,6,7,9,2,19], arr2 = [2,1,4,3,9,6]
8  输出：[2,2,2,1,4,3,3,9,6,7,19]
9
10 提示：
11      arr1.length, arr2.length <= 1000
12      0 <= arr1[i], arr2[i] <= 1000
13      arr2 中的元素 arr2[i] 各不相同
14      arr2 中的每个元素 arr2[i] 都出现在 arr1 中
15
16  */

```

分析:

- 方法一:使用map对arr2中的数据进行依次计数排序,然后把arr1中的元素按存在于arr1和不存在于arr1分别从头尾开始存放(这样arr1就被分成了两部分),然后对这两部分分别排序.

方法一:C++_map+分类排序

```

1  class solution
2  {
3      private:
4          void sort1(map<int,int>& mii ,vector<int>& vi , int left , int
   right)
5      {
6          int minIdx = 0;
7          int i = 0;
8          int j = 0;
9
10         for(i=left;i<=right;i++)
11         {
12             minIdx = i;
13             for(j=i+1;j<=right;j++)
14             {
15                 if(mii[vi[j]] < mii[vi[minIdx]])
16                 {
17                     swap(vi[j],vi[minIdx]);
18                 }
19             }
20         }
21     }
22 }
23

```

```

24 void sort2(vector<int>& vi , int left , int right)
25 {
26     int minIdx = 0;
27     int i = 0;
28     int j = 0;
29
30     for(i=left;i<=right;i++)
31     {
32         minIdx = i;
33         for(j=i+1;j<=right;j++)
34         {
35             if(vi[j] < vi[minIdx])
36             {
37                 swap(vi[j],vi[minIdx]);
38             }
39         }
40     }
41 }
42
43
44 public:
45 vector<int> relativeSortArray(vector<int>& arr1, vector<int>& arr2)
46 {
47     map<int,int> mii;
48     vector<int> ret_val(arr1.size(),0);
49     int i = 0;
50     int count1 = 0;
51     int count2 = 0;
52
53     for(i = 0 ; i < arr2.size() ; i++)
54     {
55         mii[arr2[i]] = count1++;
56     }
57
58     count1 = 0 ;
59     count2 = arr1.size() - 1 ;
60
61     for(i = 0 ; i < arr1.size() ; i++)
62     {
63         if(mii.count(arr1[i]))
64         {
65             ret_val[count1++] = arr1[i];
66         }
67         else
68         {
69             ret_val[count2--] = arr1[i];
70         }
71     }
72
73     sort1(mii,ret_val,0,count1-1);
74     sort2(ret_val,count2+1,arr1.size()-1);
75
76     return ret_val;
77
78 }
79 };
80
81 /*

```

```
82 | 执行结果:  
83 | 通过  
84 | 显示详情  
85 | 执行用时 :44 ms, 在所有 cpp 提交中击败了5.03% 的用户  
86 | 内存消耗 :8.8 MB, 在所有 cpp 提交中击败了100.00%的用户  
87 | */
```

AlimyBreak

2019.11.01