

```
1  /*
2  给你一个数组 arr ，请你将每个元素用它右边最大的元素替换，如果是最后一个元素，用 -1 替换。
3
4  完成所有替换操作后，请你返回这个数组。
5
6
7
8  示例：
9
10 输入：arr = [17,18,5,4,6,1]
11 输出：[18,6,6,6,1,-1]
12
13
14
15 提示：
16
17     1 <= arr.length <= 10^4
18     1 <= arr[i] <= 10^5
19
20 来源：力扣（LeetCode）
21 链接：https://leetcode-cn.com/problems/replace-elements-with-greatest-
    element-on-right-side
22 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
23 */
```

分析：

- 从右往左遍历即可.

方法一：C++_遍历法

```
1  class Solution
2  {
3      public:
4          vector<int> replaceElements(vector<int>& arr)
5          {
6              int length      =  arr.size()  ;
7              int max_temp1    =  0           ;
8              int max_temp2    =  0           ;
9
10
11              if(length==1)
12              {
13                  arr[0] = -1;
14              }
15              else if(length==2)
16              {
17                  arr[0] = arr[1] ;
18                  arr[1] = -1     ;
19              }
20          }
21      }
```

```

19
20     }
21     else
22     {
23         max_temp1      =  arr[length-1]
24         ;
25         max_temp2      =  arr[length-2]
26         ;
27         arr[length-1]  =  -1
28         ;
29         arr[length-2]  =  max_temp1
30         ;
31         max_temp1      =  max_temp1 > max_temp2 ? max_temp1 :
max_temp2 ;
32
33         for(int i = length - 3; i >= 0 ; i--)
34         {
35             max_temp2  =  arr[i];
36             arr[i]     =  max_temp1;
37             max_temp1  =  max_temp1 > max_temp2 ? max_temp1 :
max_temp2;
38         }
39     }
40     return arr;
41 }
42 };
43
44 /*
45 执行结果:
46 通过
47 显示详情
48 执行用时 :20 ms, 在所有 C++ 提交中击败了95.74% 的用户
49 内存消耗 :11.7 MB, 在所有 C++ 提交中击败了5.04%的用户
50 */

```