

```

1  /*
2  给你一个长度固定的整数数组 arr，请你将该数组中出现的每个零都复写一遍，并将其余的元素向右平
   移。
3  注意：请不要在超过该数组长度的位置写入元素。
4  要求：请对输入的数组 就地 进行上述修改，不要从函数返回任何东西。
5  示例 1:
6  输入: [1,0,2,3,0,4,5,0]
7  输出: null
8  解释: 调用函数后，输入的数组将被修改为: [1,0,0,2,3,0,0,4]
9  示例 2:
10 输入: [1,2,3]
11 输出: null
12 解释: 调用函数后，输入的数组将被修改为: [1,2,3]
13 提示:
14     1 <= arr.length <= 10000
15     0 <= arr[i] <= 9
16 来源：力扣（LeetCode）
17 链接：https://leetcode-cn.com/problems/duplicate-zeros
18 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。
19 */

```

分析:

- 方法一:暴力法,遍历数组遇到一个0,右移动一次.时间复杂度 $O(n * k)$, n 是数组长度, k 是0的个数.

方法一:C++_暴力法

```

1  class Solution
2  {
3
4      /* [left,right] */
5      private:
6          void move(vector<int>& arr , int left,int right)
7          {
8              int i = 0;
9              for(i=right; i > left ;i--)
10             {
11                 arr[i] = arr[i-1];
12             }
13         }
14     public:
15         void duplicateZeros(vector<int>& arr)
16         {
17             int size = arr.size();
18             for(int i = 0 ; i < size;)
19             {
20                 if(arr[i] == 0)
21                 {
22                     if(i+1 < size)

```

```
23         {
24             move(arr, i+1, size-1);
25             arr[i+1] = 0;
26         }
27         i += 2;
28     }
29     else
30     {
31         i += 1;
32     }
33 }
34 return ;
35 }
36 };
37
38 /*
39 执行结果:
40 通过
41 显示详情
42 执行用时 :76 ms, 在所有 cpp 提交中击败了25.08% 的用户
43 内存消耗 :9.2 MB, 在所有 cpp 提交中击败了100.00%的用户
44 */
```