

```

1  /*
2  给定一个二进制数组， 计算其中最大连续1的个数。
3  示例 1:
4  输入: [1,1,0,1,1,1]
5  输出: 3
6  解释: 开头的两位和最后的三位都是连续1，所以最大连续1的个数是 3。
7  注意:
8      输入的数组只包含 0 和1。
9      输入数组的长度是正整数，且不超过 10,000。
10 来源: 力扣 (LeetCode)
11 链接: https://leetcode-cn.com/problems/max-consecutive-ones
12 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。
13 */

```

分析:

- 建立两个索引(索引到指针)形成一个窗口，用来保存当前连续1形成的窗口,索引之间的距离就是窗口的宽度也就是当次连续1的长度；
- 求得连续1的窗口大小的最大值(注意边界条件)

方法一:C\_双指针法

```

1  int findMaxConsecutiveOnes(int* nums, int numsSize)
2  {
3      int i          = 0;
4      int ind_front   = 0;
5      int ind_tail    = 0;
6      int ret_val     = 0;
7
8      for(i=0;i<numsSize;i++)
9      {
10         if(nums[ind_front]==0)
11         {
12             ind_front++;
13             ind_tail++;
14         }
15         else
16         {
17             if(nums[ind_tail]==1)
18             {
19                 ind_tail++;
20             }
21             else
22             {
23                 ret_val = (ret_val>=(ind_tail-ind_front)) ? ret_val :
(ind_tail-ind_front);
24                 ind_tail++;
25                 ind_front=ind_tail;
26             }
27         }
28     }
29     ret_val = (ret_val>=(ind_tail-ind_front)) ? ret_val : (ind_tail-
ind_front);

```

```
30     return ret_val;
31 }
32
33 /*
34 执行结果:
35 通过
36 显示详情
37 执行用时 :56 ms, 在所有 C 提交中击败了95.08% 的用户
38 内存消耗 :8.7 MB, 在所有 C 提交中击败了48.76%的用户
39 */
```

---

AlimyBreak  
2019.08.23