

```

1  /*
2  我们把符合下列属性的数组 A 称作山脉：
3
4      A.length >= 3
5      存在 0 < i < A.length - 1 使得A[0] < A[1] < ... A[i-1] < A[i] > A[i+1] >
... > A[A.length - 1]
6
7  给定一个确定为山脉的数组，返回任何满足 A[0] < A[1] < ... A[i-1] < A[i] > A[i+1] >
... > A[A.length - 1] 的 i 的值。
8
9  示例 1:
10 输入: [0,1,0]
11 输出: 1
12 示例 2:
13 输入: [0,2,1,0]
14 输出: 1
15 提示:
16     3 <= A.length <= 10000
17     0 <= A[i] <= 10^6
18     A 是如上定义的山脉
19 来源：力扣（LeetCode）
20 链接：https://leetcode-cn.com/problems/peak-index-in-a-mountain-array
21 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。
22 */

```

分析:

- 按照题意，只要找出传入的“山脉数组”的顶点位置，即最大值所在位置即可。
- 首选二分法,时间复杂度为 $O(\log n)$

方法一:C\_二分法

```

1  int peakIndexInMountainArray( int* A ,
2                                int ASize
3                                )
4  {
5      int left    = 0 ;
6      int mid     = 0 ;
7      int right   = ASize - 1 ;
8
9      while(left<=right)
10     {
11         mid = left + (right - left) / 2;
12         if( (A[mid]>A[mid-1])
13             && (A[mid+1]>A[mid])
14         )
15         {
16             left = mid + 1;
17         }
18         else if( (A[mid]<A[mid-1])
19                 && (A[mid+1]<A[mid])
20             )

```

```
21     {
22         right = mid - 1;
23     }
24     else
25     {
26         return mid;
27     }
28 }
29 return -1;
30 }
31 /*
32 执行结果:
33 通过
34 显示详情
35 执行用时 :16 ms, 在所有 C 提交中击败了77.08% 的用户
36 内存消耗 :7.6 MB, 在所有 C 提交中击败了55.13%的用户
37 */
```

---

AlimyBreak  
2019.09.07