

852. Peak Index in a Mountain Array

Let's call an array `arr` a **mountain** if the following properties hold:

- `arr.length >= 3`
- There exists some `i` with `0 < i < arr.length - 1` such that:
 - `arr[0] < arr[1] < ... arr[i-1] < arr[i]`
 - `arr[i] > arr[i+1] > ... > arr[arr.length - 1]`

Given an integer array `arr` that is **guaranteed** to be a mountain, return any `i` such that `arr[0] < arr[1] < ... arr[i - 1] < arr[i] > arr[i + 1] > ... > arr[arr.length - 1]`.

Example 1:

Input: `arr = [0,1,0]`

Output: 1

Example 2:

Input: `arr = [0,2,1,0]`

Output: 1

Example 3:

Input: `arr = [0,10,5,2]`

Output: 1

Example 4:

Input: `arr = [3,4,5,1]`

Output: 2

Example 5:

Input: `arr = [24,69,100,99,79,78,67,36,26,19]`

Output: 2

```
def peakIndexInMountainArray(self, arr: List[int]) -> int:
    return arr.index(max(arr))

    #Second Approach(logN)
    # lo = 0
```

```
# n = len(arr)
# hi = n-1
# while lo<=hi:
#     mid = lo+(hi-lo)//2
#     prev = (mid-1+n)%n
#     nextt = (mid+1+n)%n
#     if arr[mid] > arr[nextt] and arr[mid] > arr[prev]:
#         return mid
#     elif arr[mid] > arr[mid+1]:
#         hi = mid-1
#     elif arr[mid] < arr[mid+1]:
#         lo = mid +1
```