**Problem #852: Peak Index in a Mountain Array (Medium)**

<https://leetcode.com/problems/peak-index-in-a-mountain-array/description/>

**My Solution:**

1. Let n be length of arr.
2. If n is less than 3, then return
3. If n is equal to 3, then return the index of the middle element, which is 1.
4. Let first be 0 and last be (n – 1).
5. While first is less than last, middle is the average of first and last obtained by dividing the sum of first and last by 2 (integer division)

If the element at middle index is greater than the elements on both sides, then return middle.

If the element at middle index is greater than the previous one but less than the next element, the, ascending values. So the peak is in the second half and therefore assign middle to first.

Otherwise, assign middle to last.

class Solution:

def peakIndexInMountainArray(self, arr: List[int]) -> int:

n = len(arr)

if n < 3:

return

elif n == 3:

return 1

first = 0

last = n - 1

while first < last:

middle = (first + last) // 2

if arr[middle] > arr[middle + 1] and arr[middle] > arr[middle - 1]:

return middle

elif arr[middle - 1] < arr[middle] and arr[middle] < arr[middle + 1]:

first = middle

else:

last = middle