Problem #153: Find Minimum in a Sorted Rotated Array

<https://leetcode.com/problems/find-minimum-in-rotated-sorted-array/description/>

**My Solution:**

1. Let n be the length of nums.
2. Let first be 0
3. Let last be (n – 1)
4. While first is less than last, let middle be the average of first and last.

If nums at index middle is less than nums at index last, then assign middle to last.

Otherwise assign (middle + 1) to first.

1. Finally return nums at index first.

class Solution:

def findMin(self, nums: List[int]) -> int:

n = len(nums)

first = 0

last = n - 1

while first < last:

middle = (first + last)//2

if nums[middle] < nums[last]:

last = middle

else:

first = middle + 1

return nums[first]