**Problem #452: Minimum Number of Arrows to Burst Balloon (Medium)**

<https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons/description/>

**My Solution:**

1. Sort nums array and put it in nums.
2. Let n be the length of nums
3. If n is odd, then median is nums at index n divided by 2 (integer division).
4. If n is even, then meidan is nums at index n divided by 2 and then subtract 1.
5. Initialize moves to 0.
6. Iterate through nums array. For each num in nums, get the absolute difference of nums and median and add this to moves.
7. Return moves.

class Solution:

def minMoves2(self, nums: List[int]) -> int:

nums = sorted(nums)

n = len(nums)

if n % 2 == 1:

median = nums[n//2]

else:

median = nums[(n//2) - 1]

moves = 0

for num in nums:

moves += abs(num - median)

return moves