Even or Odd Sum Based on Position

Question:

Given an array of integers, return 'even' if the sum of numbers at even positions is greater than the sum at odd positions. Return 'odd' if the sum at odd positions is greater than even positions. If both are equal, return 'equal'.

Python Code:

```
def even_or_odd(arr):
      even_sum = sum(arr[i] for i in range(0, len(arr), 2))   # Sum of even-indexed elements
    odd_sum = sum(arr[i] for i in range(1, len(arr), 2))   # Sum of odd-indexed elements

    if even_sum > odd_sum:
        return "even"
    elif odd_sum > even_sum:
        return "odd"
    else:
        return "equal"

# Example usage:
arr = [1, 2, 3, 4, 5, 6]
print(even_or_odd(arr))  # Output: "equal"
```

Explanation:

The function iterates through the given list, calculating the sum of elements at even indices and odd indices separately. It then compares these sums:
- If even sum is greater, it returns 'even'.
- If odd sum is greater, it returns 'odd'.
- If both sums are equal, it returns 'equal'.

Time Complexity:

The function iterates through the list twice: once for even indices and once for odd indices. Each iteration takes O(n/2) time, so the overall time complexity is:

O(n)

Space Complexity:

The function uses a constant amount of extra space (two integer variables for sums), so the space complexity is:

O(1)