```python
def find_min_max(arr):
    min_val = max_val = arr[0]
    for num in arr[1:]:
        min_val = min(min_val, num)
        max_val = max(max_val, num)
    return min_val, max_val

# Test cases
test_cases = [
    [8, [5, 7, 3, 4, 9, 12, 6, 2]],
    [9, [1, 3, 5, 7, 9, 11, 13, 15, 17]],
    [10, [22, 34, 35, 36, 43, 67, 12, 13, 15, 17]]
]

for n, arr in test_cases:
    min_val, max_val = find_min_max(arr)
    print(f"Min = {min_val}, Max = {max_val}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
Min = 2, Max = 12
Min = 1, Max = 17
Min = 12, Max = 67

[Done] exited with code=0 in 0.176 seconds
```

```python
def find_min_max_sorted(arr):
    return arr[0], arr[-1]

# Test cases
test_cases = [
    [8, [2, 4, 6, 8, 10, 12, 14, 18]],
    [9, [11, 13, 15, 17, 19, 21, 23, 35, 37]],
    [10, [22, 34, 35, 36, 43, 67, 12, 13, 15, 17]]
]

for n, arr in test_cases:
    min_val, max_val = find_min_max_sorted(arr)
    print(f"Min = {min_val}, Max = {max_val}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
Min = 2, Max = 18
Min = 11, Max = 37
Min = 22, Max = 17
```

```python
def merge_sort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    left_half = merge_sort(arr[:mid])
    right_half = merge_sort(arr[mid:])

    return merge(left_half, right_half)

def merge(left, right):
    merged = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            merged.append(left[i])
            i += 1
        else:
            merged.append(right[j])
            j += 1

    merged.extend(left[i:])
    merged.extend(right[j:])

    return merged

# Test cases
test_cases = [
    [8, [31, 23, 35, 27, 11, 21, 15, 28]],
    [10, [22, 34, 25, 36, 43, 67, 52, 13, 65, 17]]
]

for n, arr in test_cases:
    sorted_arr = merge_sort(arr)
    print(sorted_arr)
```

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[11, 15, 21, 23, 27, 28, 31, 35]
[13, 17, 22, 25, 34, 36, 43, 52, 65, 67]

[Done] exited with code=0 in 0.148 seconds
```

```python
def merge_sort(arr):
    comparison_count = [0]
    def merge_sort_helper(array):
        if len(array) > 1:
            mid = len(array) // 2
            left_half = array[:mid]
            right_half = array[mid:]
            merge_sort_helper(left_half)
            merge_sort_helper(right_half)
            i = j = k = 0
            while i < len(left_half) and j < len(right_half):
                comparison_count[0] += 1
                if left_half[i] < right_half[j]:
                    array[k] = left_half[i]
                    i += 1
                else:
                    array[k] = right_half[j]
                    j += 1
                k += 1
            while i < len(left_half):
                array[k] = left_half[i]
                i += 1
                k += 1
            while j < len(right_half):
                array[k] = right_half[j]
                j += 1
                k += 1
    merge_sort_helper(arr)
    return arr, comparison_count[0]
```

(variable) right_half: Any

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[11, 15, 21, 23, 27, 28, 31, 35]
[13, 17, 22, 25, 34, 36, 43, 52, 65, 67]
```

```python
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[0]
    less_than_pivot = [x for x in arr[1:] if x <= pivot]
    greater_than_pivot = [x for x in arr[1:] if x > pivot]
    return quick_sort(less_than_pivot) + [pivot] + quick_sort(greater_than_p

# Test Case 1
arr1 = [10, 16, 8, 12, 15, 6, 3, 9, 5]
print(quick_sort(arr1))  # Expected Output: [3, 5, 6, 8, 9, 10, 12, 15, 16]

# Test Case 2
arr2 = [12, 4, 78, 23, 45, 67, 89, 1]
print(quick_sort(arr2))  # Expected Output: [1, 4, 12, 23, 45, 67, 78, 89]

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
[3, 5, 6, 8, 9, 10, 12, 15, 16]
[1, 4, 12, 23, 45, 67, 78, 89]

[Done] exited with code=0 in 0.172 seconds
```

```python
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    less_than_pivot = [x for x in arr if x < pivot]
    equal_to_pivot = [x for x in arr if x == pivot]
    greater_than_pivot = [x for x in arr if x > pivot]
    return quick_sort(less_than_pivot) + equal_to_pivot + quick_sort(gr

# Test Case
arr = [19, 72, 35, 46, 58, 91, 22, 31]
print(quick_sort(arr))  # Expected Output: [19, 22, 31, 35, 46, 58, 72,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[19, 22, 31, 35, 46, 58, 72, 91]

[Done] exited with code=0 in 0.15 seconds
```

```python
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    comparisons = 0

    while left <= right:
        mid = (left + right) // 2
        comparisons += 1
        if arr[mid] == target:
            return mid, comparisons
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return -1, comparisons

# Test Case 1
arr1 = [5, 10, 15, 20, 25, 30, 35, 40,
result, count = binary_search(arr1, 20)
print(f"Index: {result}, Comparisons: {count}")  # Expected Output: Index: 3
```

(variable) count: int

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
Index: 3, Comparisons: 4

[Done] exited with code=0 in 0.124 seconds
```

```python
def binary_search(arr, target):
    left, right = 0, len(arr) - 1

    while left <= right:
        mid = (left + right) // 2
        print(f"Midpoint: {mid}, Value: {arr[mid]}")
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return -1

# Test Case
arr = [3, 9, 14, 19, 25, 31, 42, 47, 53]
print(binary_search(arr, 31))  # Expected Output: 5

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
Index: 3, Comparisons: 4

[Done] exited with code=0 in 0.124 seconds
```

```python
import heapq

def k_closest(points, k):
    return heapq.nsmallest(k, points, key=lambda x: x[0]**2 + x[1]**2)

# Test Cases
print(k_closest([[1,3],[-2,2],[5,8],[0,1]], 2))  # Expected Output: [[-2, 2]
print(k_closest([[1, 3], [-2, 2]], 1))  # Expected Output: [[-2, 2]]
print(k_closest([[3, 3], [5, -1], [-2, 4]], 2))  # Expected Output: [[3, 3],

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[[0, 1], [-2, 2]]
[[-2, 2]]
[[3, 3], [-2, 4]]

[Done] exited with code=0 in 0.125 seconds
```

```python
from collections import Counter

def four_sum_count(A, B, C, D):
    AB_sum = Counter(a + b for a in A for b in B)
    return sum(AB_sum[-c - d] for c in C for d in D)

# Test Cases
print(four_sum_count([1, 2], [-2, -1], [-1, 2], [0, 2]))  # Expected Output:
print(four_sum_count([0], [0], [0], [0]))  # Expected Output: 1

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
2
1

[Done] exited with code=0 in 0.18 seconds
```