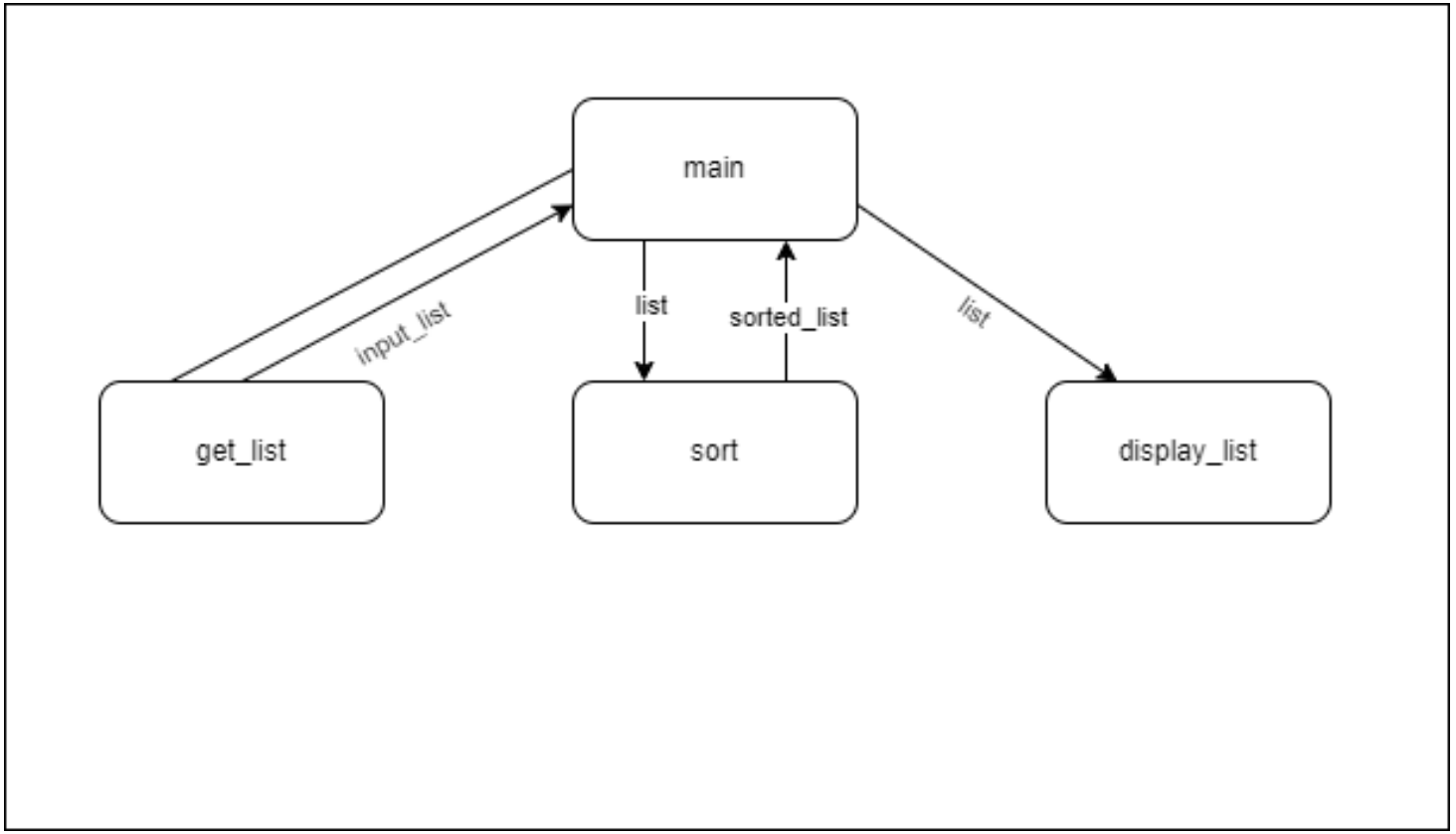
Lab 12:

Structure Chart



Pseudocode

```
1. FUNCTION sort_rec(array, i_start, i_end):
2.     i_up = i_start
3.     i_down = i_end
4.     i_pivot = (i_start + i_end) // 2
5.
6.     # End condition
7.     IF i_up >= i_down or array == []:
8.         RETURN array
9.
10.       WHILE i_up < i_down:
11.
12.           WHILE array[i_up] <= array[i_pivot] and i_up < i_pivot:
13.               i_up += 1
14.
15.           WHILE array[i_down] >= array[i_pivot] and i_down > i_pivot:
16.               i_down -= 1
17.
18.           IF i_pivot == i_up:
19.               i_pivot = i_down
20.           ELSE IF i_pivot == i_down:
21.               i_pivot = i_up
22.
23.           Swap array[i_up] and array[i_down]
24.
25.       sort_rec(array, i_start, i_pivot - 1) # left sort
26.       sort_rec(array, i_pivot + 1, i_end) # right sort
27.       RETURN array
```

Trace

Line	array	i_start	i_end	i_up	i_down	i_pivot	array[i_up]	array[i_down]	array[i_pivot]
1	[2, 18, 4]	0	2	/	/	/	/	/	/
2	[2, 18, 4]	0	2	0	/	/	2	/	/
3	[2, 18, 4]	0	2	0	2	/	2	4	/
4	[2, 18, 4]	0	2	0	2	1	2	4	18
7	[2, 18, 4]	0	2	0	2	1	2	4	18
10	[2, 18, 4]	0	2	0	2	1	2	4	18
12	[2, 18, 4]	0	2	0	2	1	2	4	18
13	[2, 18, 4]	0	2	1	2	1	18	4	18
12	[2, 18, 4]	0	2	1	2	1	18	4	18
15	[2, 18, 4]	0	2	1	2	1	18	4	18
18	[2, 18, 4]	0	2	1	2	1	18	4	18
19	[2, 18, 4]	0	2	1	2	2	18	4	4
23	[2, 4, 18]	0	2	1	2	2	18	4	4
25	[2, 4, 18]	0	2	0	2	1	2	18	4
26	[2, 4, 18]	1	2	1	2	2	4	18	18
27	[2, 4, 18]	1	2	1	2	2	4	18	18

Modularization Metrics

Cohesion:

- FUNCTION find_sorted_subarray
    - The cohesion is strong because the function does exactly what it should. There aren't any extraneous or extra code.

Coupling:

- FUNCTION find_sorted_subarray
    - The function has weak or low coupling. It does not depend on any other functions.

Efficiency

```
FUNCTION sort_rec(array, i_start, i_end):                          # O(1)
    i_up = i_start                                                 # O(1)
    i_down = i_end                                                 # O(1)
    i_pivot = (i_start + i_end) // 2                               # O(1)

    # End condition
    IF i_up >= i_down or array == []:                              # O(1)
        RETURN array                                               # O(1)

    WHILE i_up < i_down:                                           # O(n)

        WHILE array[i_up] <= array[i_pivot] and i_up < i_pivot:    # O(n)
            i_up += 1                                              # O(1)
        WHILE array[i_down] >= array[i_pivot] and i_down > i_pivot: # O(n)
```

```
            i_down -= 1                                         # O(1)

        IF i_pivot == i_up:                                     # O(1)
            i_pivot = i_down                                    # O(1)
        ELSE IF i_pivot == i_down:                              # O(1)
            i_pivot = i_up                                      # O(1)

        SWAP array[i_up] and array[i_down]                      # O(1)
    sort_rec(array, i_start, i_pivot - 1)                       # O(log n)
    sort_rec(array, i_pivot + 1, i_end)                         # O(log n)
    RETURN array                                                # O(1)
```

Overall Efficiency is: O(n log n)

Test Cases

```
test_inputs = [
    [2, 18, 4],         # Trace example Unsorted
    [3, 2, 1, 5, 4],     # Normal Unsorted
    [1, 2, 3, 4, 5],     # Already Sorted
    [5, 4, 3, 2, 1],     # Reverse Sorted
    [1],                 # Single Element
    [],                  # Empty Array
    [2, 2, 2],     # All elements the same
    ["Bethlehem", "Nativity", "Magi", "Emmanuel", "Baby Jesus", "Angels", "Shepherds"] # Array of Strings Unsorted
]
expected_outputs = [
    [2, 4, 18],
    [1, 2, 3, 4, 5],
    [1, 2, 3, 4, 5],
    [1, 2, 3, 4, 5],
    [1],
    [],
    [2, 2, 2],
    ["Angels", "Baby Jesus", "Bethlehem", "Emmanuel", "Magi", "Nativity", "Shepherds"]
]
```

Submission Notes

- How long did it take for you to complete this assignment?
    - 5 hours
- What was the hardest part of the assignment?
    - The hardest part was doing the trace and figuring out the efficiency
- Was there anything unclear about the instructions or how you were to complete this lab?
    - Everything was very clear.