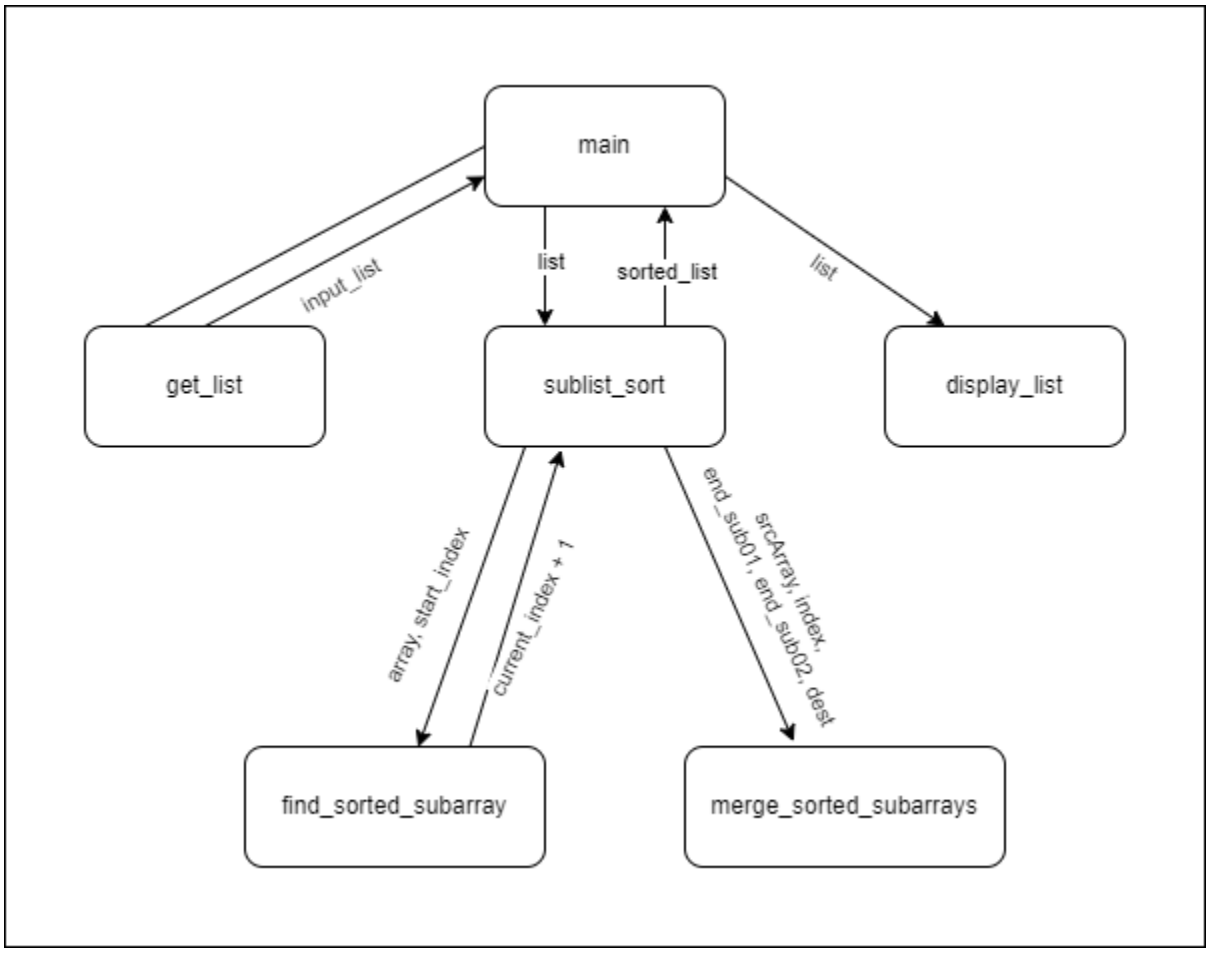
Structure Chart:



Pseudocode:

```
FUNCTION find_sorted_subarray(array, start_index)
    current_index = start_index
    WHILE current_index < (length(array - 1)) AND array[current_index] <=
array[current_index + 1]
        current_index += 1
    RETURN current_index + 1
FUNCTION END

FUNCTION merge_sorted_subarrays(sourceArray, start_index, end_of_subarray01,
end_of_subarray02, destination)
     current_index_in_sub01 = start_index
     current_index_in_sub02 = end_of_subarray01
     destination_index = start_index

    WHILE current_index_in_sub01 < end_of_subarray01 AND current_index_in_sub02 <
end_of_subarray02
        IF sourceArray[current_index_in_sub01] <= sourceArray[current_index_in_sub02]
```

```
            destination[destination_index] = sourceArray[current_index_in_sub01]
            current_index_in_sub01 += 1
        ELSE
            destination[destination_index] = sourceArray[current_index_in_sub02]
            current_index_in_sub02 += 1
        destination_index += 1

    WHILE current_index_in_sub01 < end_of_subarray01
        destination[destination_index] = sourceArray[current_index_in_sub01]
        current_index_in_sub01 AND destination_index += 1

    WHILE current_index_in_sub02 < end_of_subarray02
        destination[destination_index] = sourceArray[current_index_in_sub02]
        current_index_in_sub02 AND destination_index += 1
FUNCTION END


FUNCTION sublist_sort(array)
     temp_array = Array equal to an array equal to the size of the original array
     array_size = length(array)
     is_sorted = FALSE

    WHILE NOT is_sorted
         is_sorted = TRUE
         start_index = 0

        WHILE start_index < array_size
             end_of_subarray01 = find_sorted_subarray(array, start_index)
             end_of_subarray02 = find_sorted_subarray(array, end_of_subarray01)

            IF end_of_subarray02 > array_size
                 end_of_subarray02 = array_size

            merge_sorted_subarrays(array, start_index, end_of_subarray01,
end_of_subarray02, temp_array)
            IF end_of_subarray02 != array_size
                 is_sorted = FALSE

             start_index = end_of_subarray02

        SWAP array with temp_array

    RETURN array
FUNCTION END
```

Submission Notes:

- How long did it take for you to complete this assignment?
    - 16 Hours at least. I spent a lot of time on this this week and it was hard to track. But I spent several hours every day this week.
- What was the hardest part of the assignment?
    - The whole thing was hard. The hardest part was by far the merge. Figuring out the sort was hard but once I switched from using for loops to while loops that fixed a lot of problems. I did a lot of whiteboard with Brother Godderidge, Brother Helfrich, and Isaac Huffman, one of my classmates. Through that I was able to figure out the sort but the merge was super difficult to work out and think through.
- Was there anything unclear about the instructions or how you were to complete this lab?
    - The algorithm was confusing until I saw it done in class and with the video. Then the lab made sense when I read it again.