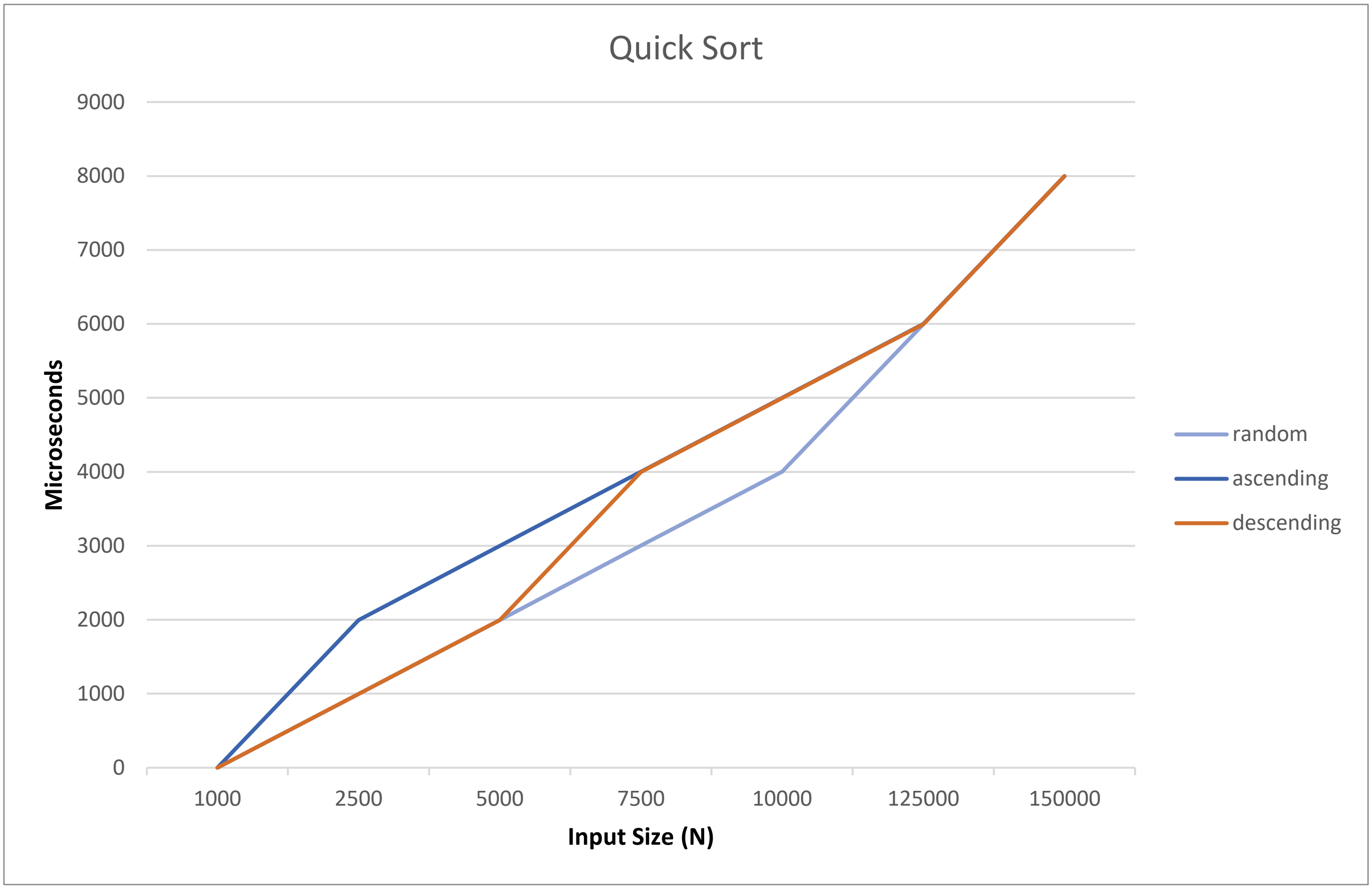
CIS 22C: Team Project Report

Group 7:

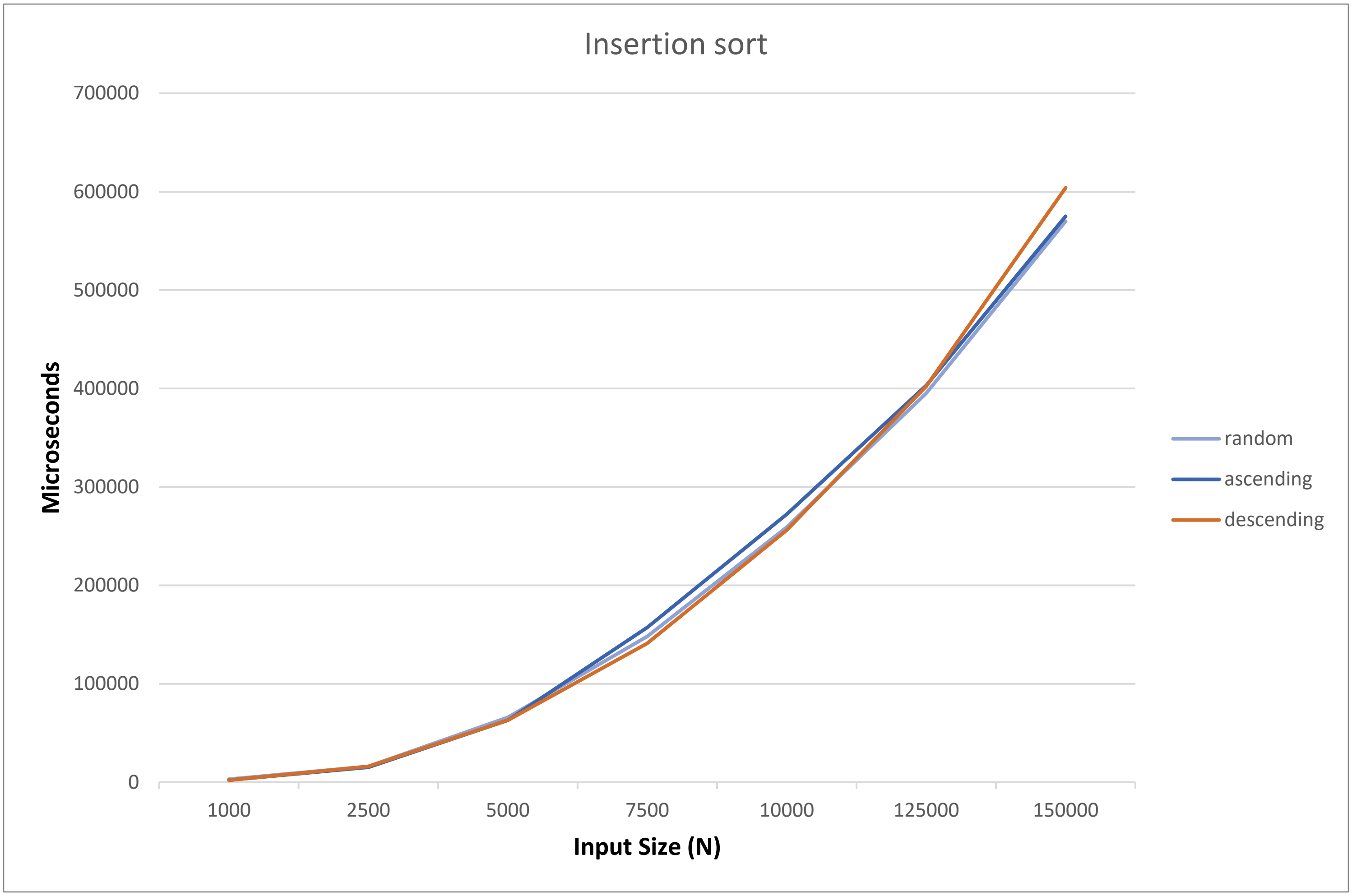
- Adarsh B Shankar
- Kurmanbek Karaev
- Tai Le
- Michael Pham
- Michelle Pham

- ## Non – Recursive Quicksort:



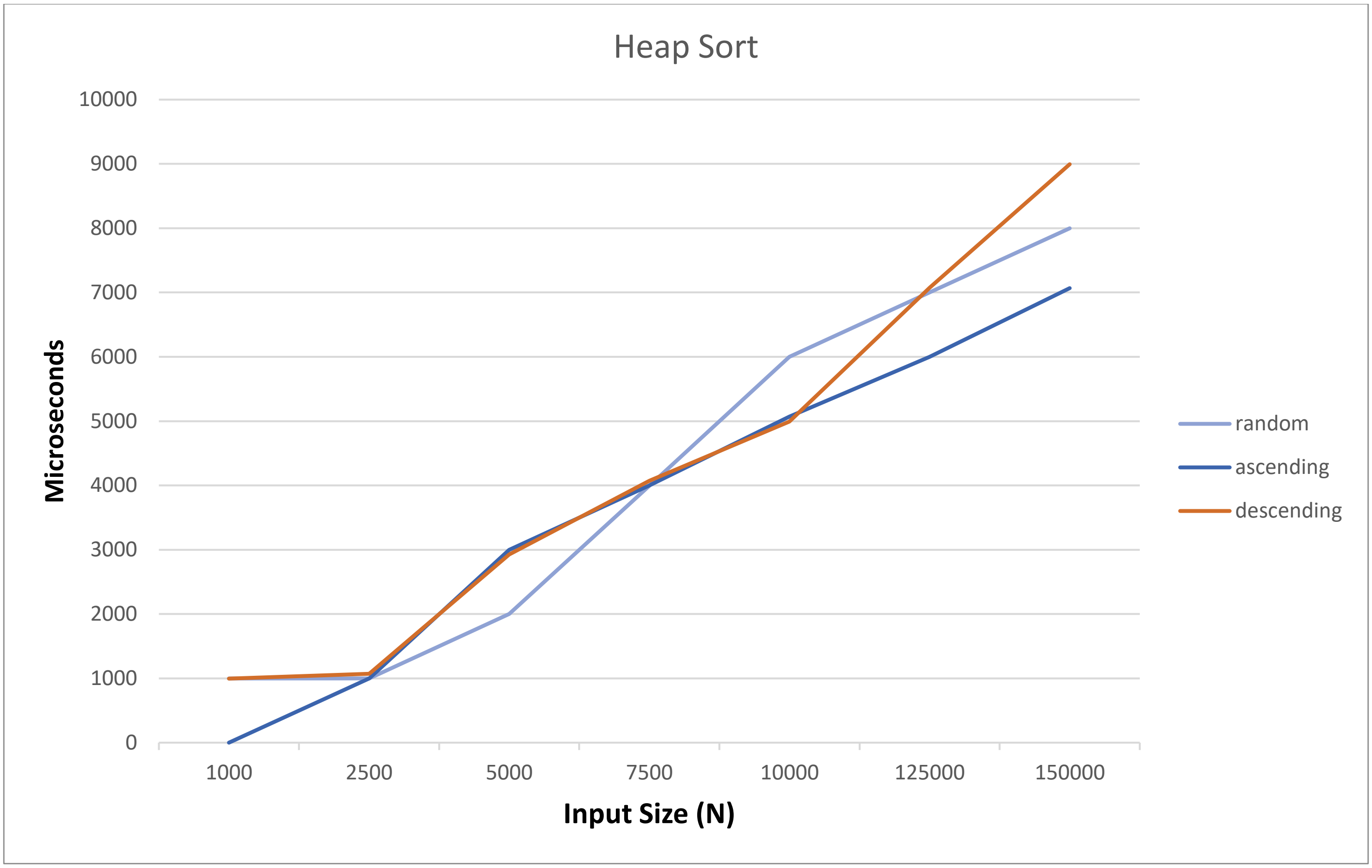
- Best Case Scenario: O(N)
- Worst Case Scenario: O(N^2)
- Average Case Scenario: O(N log N)

- ## Insertion Sort:



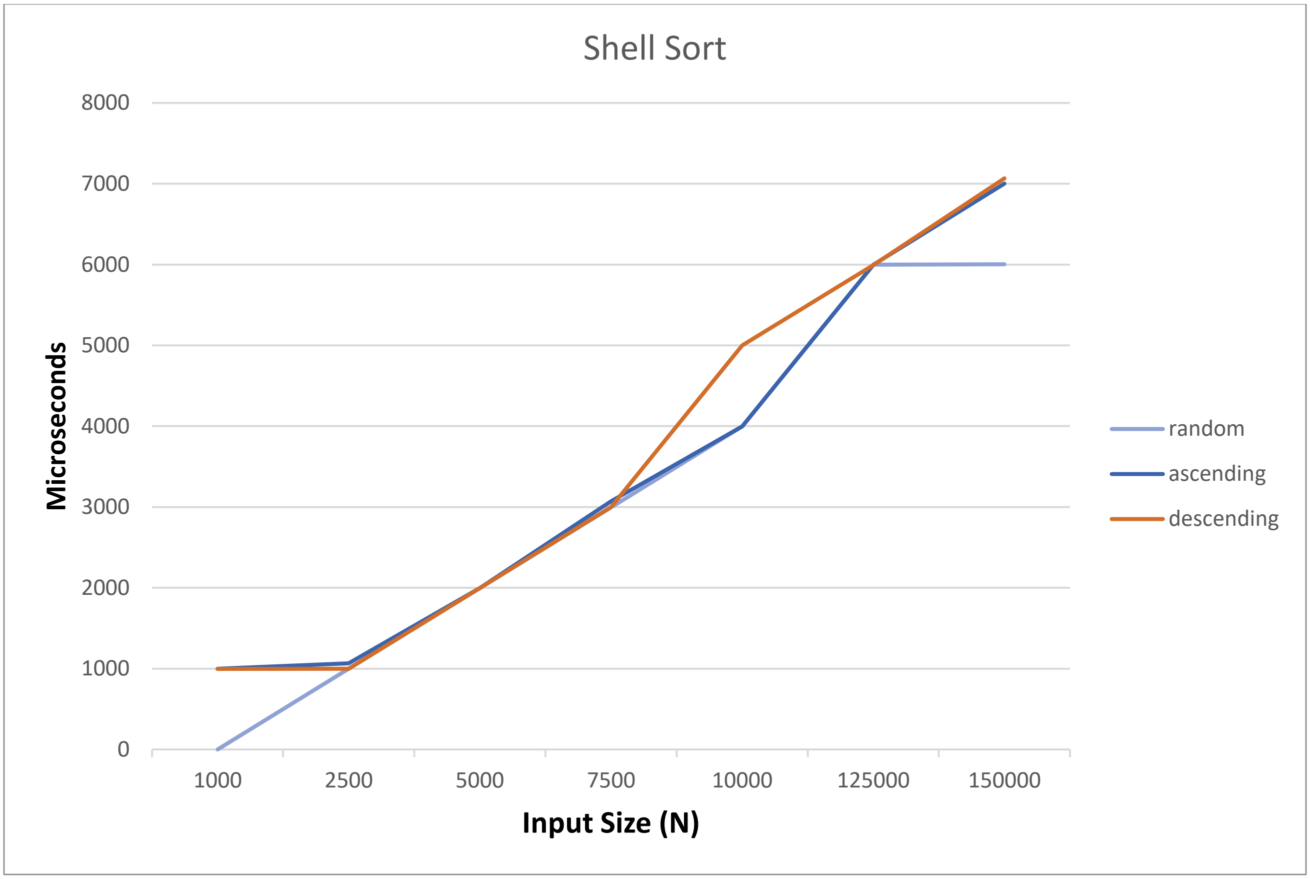
- Best Case Scenario: O(N)
- Worst Case Scenario: O(N^2)
- Average Case Scenario: O(N^2)

- ## Heapsort:



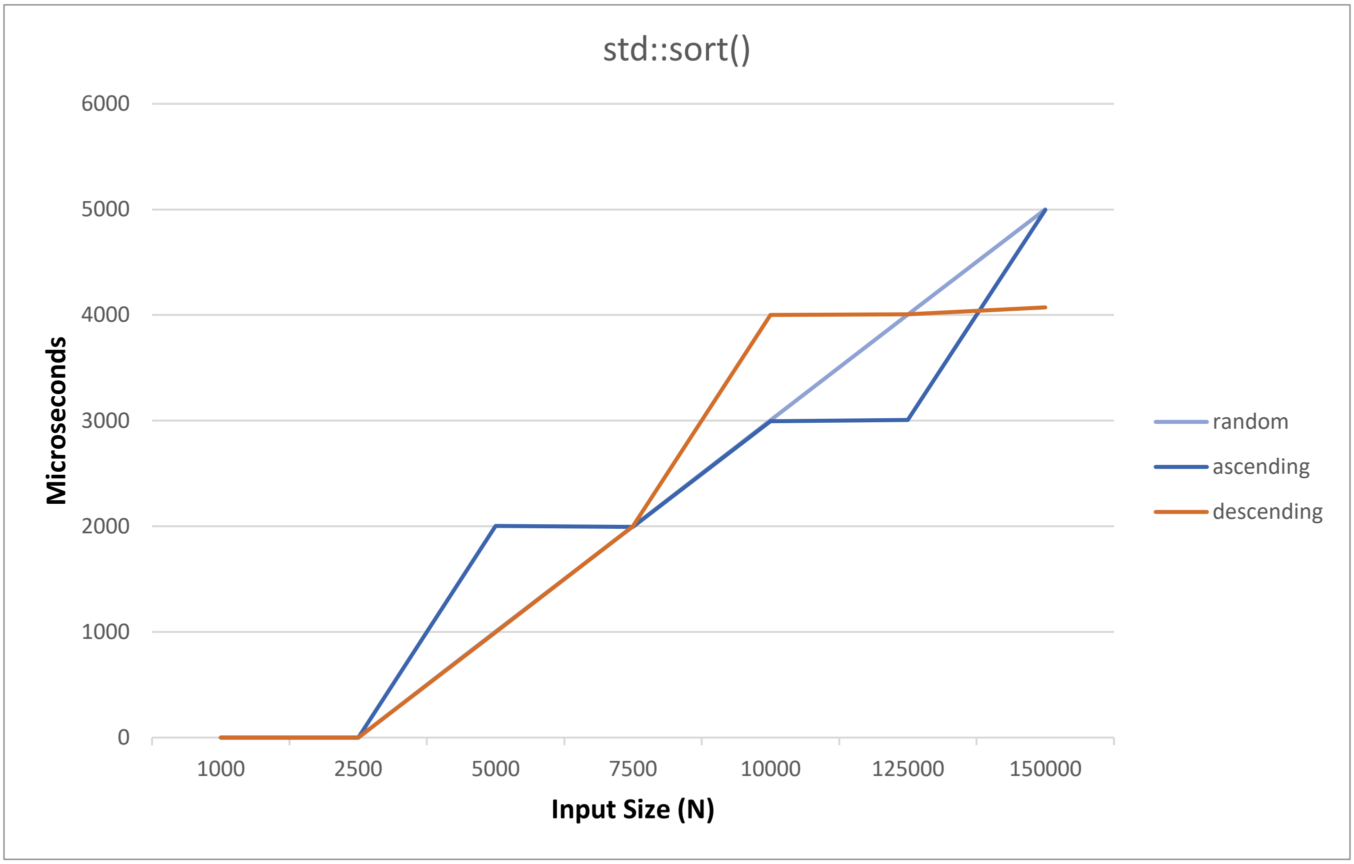
- Best Case Scenario: O(N log N) or O(N)
- Worst Case Scenario: O(N log N)
- Average Case Scenario: O(N log N)

- ## Shellsort:



- Best Case Scenario: O(N log N)
- Worst Case Scenario: O(N^2)
- Average Case Scenario: depends on the gap sequence

- Std::Sort():



Conclusion:

Comparing the non-recursive Quicksort method and the std::sort() method, we see that the std::sort() method is quicker but only by a slight margin. This is maybe because the std::sort() method uses the recursive implementation of quick sort.

Looking at the graph for Insertion Sort we can see that it is the most consistent regardless of the size and arrangement of input, but it does take the longest time out of all the other methods.

Heapsort and Shellsort are also viable and efficient sorting algorithms even for larger inputs as they both have almost the same time complexities. But clearly out of all the sorting algorithms Quicksort is by far the fastest excluding the std::sort() method.