**Task 1**

**Written task for chosen sorting algorithm in (C)**

1. Specify the sorting algorithm chosen for (c) and state its asymptotic speed in terms of the Best, Average and Worst Case.
   * The CustomSort Class uses the built in sorting algorithm of C#.
   * In terms of asymptotic speeds:
     1. Best : O(n log n)
     2. Average : O(n log n)
     3. Worst : O(n log n)

The above avoids the worst case scenario entirely and always remains within the bounds of O(n log n).

1. Compare the asymptotic speeds above to those of the other 2 sorting algorithms implemented i.e. Merge Sort and Quick Sort and state which algorithm, would be the most ideal one for sorting objects.
   * Merge Sort Asymptotic Speeds:
     1. Best : O(n log n)
     2. Average : O(n log n)
     3. Worst : O(n log n)
   * Quick Sort Asymptotic Speeds:
     1. Best : O(n log n)
     2. Average : O(n log n)
     3. Worst : O(n²)

As for the most ideal algorithm, it would be CustomSort as it is highly optimized and uses the least memory.[1]

**Task 2**

Written task for PRNG correctness and intractability.

1. Is your PRNG implementation correct?

*Base your answer on the results obtained for code requested in point A.*

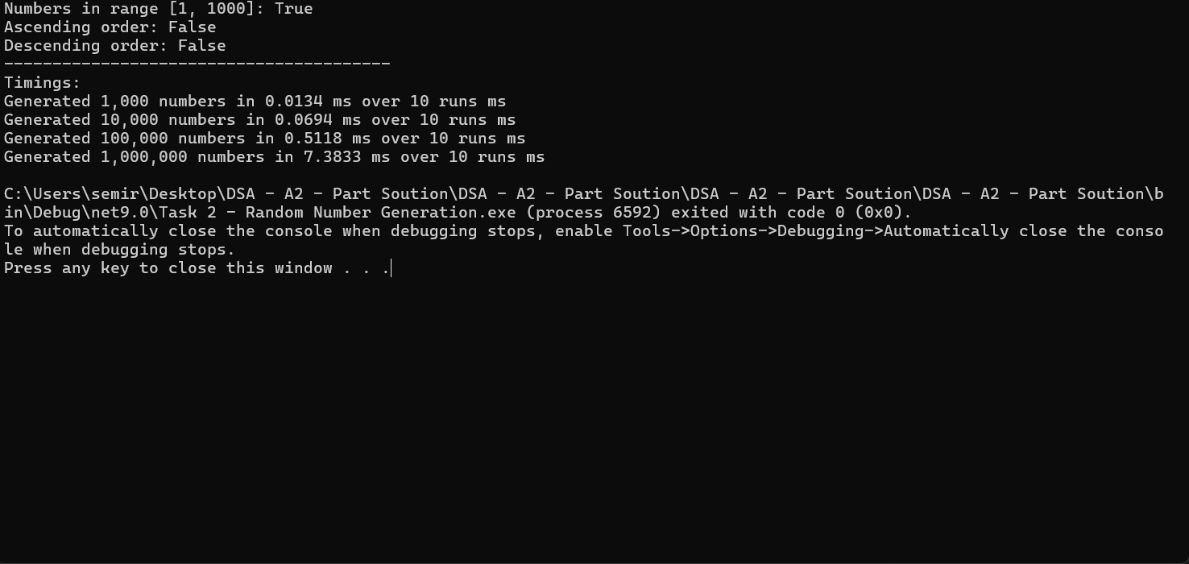
* + Yes, as shown in the below screenshot (Results), the numbers generated are within the range given and aren’t in descending or ascending order.

1. Is your PRNG implementation intractable?

*Base your answer on the Log-Log Graph produced in point B.*

* + No, although the timings produced initially gave results close to 0 due to taking the time with milliseconds instead of microseconds, by taking the average of 10 tests, a consistent value could be obtained to be plotted.
  + Time grows in a linear way as the size of the test increases. Hence, the graph shows a line with a slope.

Results:



Log-Log Graph:

A graph with a line and a point

AI-generated content may be incorrect.

FlowChart:

A diagram of a program

AI-generated content may be incorrect.

**References:**

[1] Dotnet-Bot, “List.Sort Method (System.Collections.Generic),” *Microsoft Learn*. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1.sort?view=net-9.0>. [Accessed: Jun. 7, 2025].