Task 1 – WRITTEN TASK FOR CHOSEN SORTING ALGORITHM IN (C)

Asymptotic Rates of Heap Sort:

* Best Case: O(n log n)
* Average Case: O(n log n)
* Worst Case: O(n log n)

Comparison with Merge Sort and Quick Sort:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Best Case | Average Case | Worst Case |
| Heap Sort | O(n log n) | O(n log n) | O(n log n) |
| Merge Sort | O(n log n) | O(n log n) | O(n log n) |
| Quick Sort | O(n log n) | O(n log n) | O(n²) |

Most Ideal Algorithm for Sorting Objects:

In order to arrange objects by delivery date, Merge Sort would be the most appropriate option since:

It is of O(n log n) in the best, average, and worst cases.

It is a balanced sort which keeps a relative order of equal elements, which is often required when sorting objects

Unlike Quick Sort, it doesn't have a worst-case O(n²) scenario Although Heap Sort is also O(n log n), it is not fast and usually has higher constant factors than Merge Sort.

Task 2 – WRITTEN TASK FOR PRNG CORRECTNESS AND INTRACTABILITY

1. Yes, the implementation is correct because:
   1. The validation ranges are between 1 to 1000
   2. Numbers generated are neither sorted to Ascending or Descending which indicates randomness.

These match with the expected behaviors of the algorithm showing that it is correct.

1. The PRNG implementation is correct due:
   1. The log-log graph timing results show the following relationship between input size and time.

|  |  |
| --- | --- |
| Size | Time |
| 1000 | 75 |
| 10000 | 705 |
| 100000 | 10674 |
| 1000000 | 85558 |

* 1. Time Complexity scales linearly therefore the input size (O(n)), as each additional number requires a constant time operation. This is proved by:
     1. 10x increases in input size and so does time.

A black screen with white text

AI-generated content may be incorrect.