**Assignment 2 – Report**

**Data Structure & Algorithms**

**Dat Duong – 100886108**

**­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Introduction:**

This assignment aims to provide a comprehensive understanding of sorting algorithms by visually and audibly representing their execution. It is crucial for gaining insights into how sorting algorithms work, their time complexities, and how different data structures impact sorting efficiency.

1. **Detailed Visualization:**
2. **Sorting Processes:**

* **Merge Sort:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 1 | 30 | 2 | 51 | 6 | 29 | 7 | 67 | 15 | 118 | 4 | 89 | 23 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | 1 | 30 | 2 | 51 | 6 | 29 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7 | 67 | 15 | 118 | 4 | 89 | 23 |

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | 1 | 30 | 2 |

|  |  |  |
| --- | --- | --- |
| 51 | 6 | 29 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 67 | 15 | 118 |

|  |  |  |
| --- | --- | --- |
| 4 | 89 | 23 |

|  |  |
| --- | --- |
| 11 | 1 |

|  |  |
| --- | --- |
| 30 | 2 |

|  |  |
| --- | --- |
| 51 | 6 |

|  |
| --- |
| 29 |

|  |  |
| --- | --- |
| 7 | 67 |

|  |  |
| --- | --- |
| 15 | 118 |

|  |  |
| --- | --- |
| 4 | 89 |

|  |
| --- |
| 23 |

|  |
| --- |
| 11 |

|  |
| --- |
| 1 |

|  |
| --- |
| 30 |

|  |
| --- |
| 2 |

|  |
| --- |
| 51 |

|  |
| --- |
| 6 |

|  |
| --- |
| 29 |

|  |
| --- |
| 7 |

|  |
| --- |
| 67 |

|  |
| --- |
| 15 |

|  |
| --- |
| 118 |

|  |
| --- |
| 4 |

|  |
| --- |
| 89 |

|  |
| --- |
| 23 |

|  |  |
| --- | --- |
| 1 | 11 |

|  |  |
| --- | --- |
| 2 | 30 |

|  |  |
| --- | --- |
| 6 | 51 |

|  |
| --- |
| 29 |

|  |  |
| --- | --- |
| 7 | 67 |

|  |  |
| --- | --- |
| 15 | 118 |

|  |  |
| --- | --- |
| 4 | 89 |

|  |
| --- |
| 23 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 11 | 30 |

|  |  |  |
| --- | --- | --- |
| 6 | 29 | 51 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 15 | 67 | 118 |

|  |  |  |
| --- | --- | --- |
| 4 | 23 | 89 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | 11 | 29 | 30 | 51 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | 7 | 15 | 23 | 67 | 89 | 118 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 4 | 6 | 7 | 11 | 15 | 23 | 29 | 30 | 51 | 67 | 89 | 118 |

* **Quick Sort:**

pivot**,** itemfromLeft, itemfromRight

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 1 | 30 | 2 | 51 | 6 | 29 | 7 | 67 | 15 | 118 | 4 | 89 | 23 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 30 | 2 | 51 | 6 | 29 | 7 | 67 | 15 | 118 | 4 | 89 | 23 |

|  |
| --- |
| 11 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 4 | 2 | 51 | 6 | 29 | 7 | 67 | 15 | 118 | 30 | 89 | 23 |

|  |
| --- |
| 11 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 4 | 2 | 7 | 6 | 29 | 51 | 67 | 15 | 118 | 30 | 89 | 23 |

|  |
| --- |
| 11 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 4 | 2 | 7 | 6 | 11 | 51 | 67 | 15 | 118 | 30 | 89 | 23 | 29 |

1. **Comparison Table:**

|  |  |  |
| --- | --- | --- |
|  | Merge Sort | Quick Sort |
| Best Case | Ω (n log(n)) | Ω (n log(n)) |
| Average Cash | Θ (n log(n)) | Θ (n log(n)) |
| Worst Case | O (n log(n)) | O (n^2) |

1. **Code Implementation:**

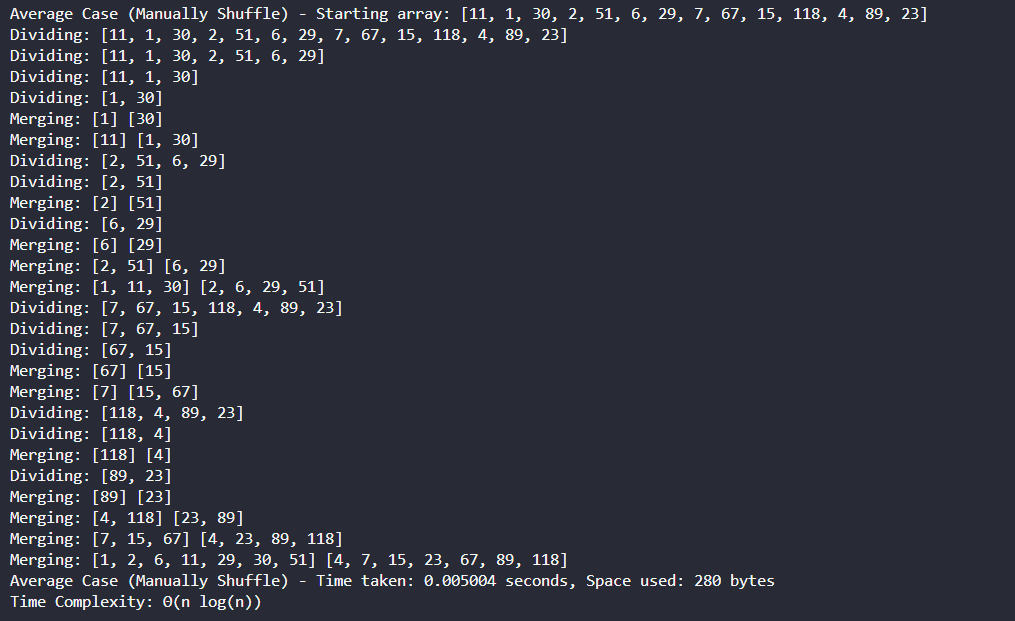
* GitHub: <https://github.com/datduong1205/Sorting-Algorithm-Visualization-and-Simulation>

**Code Snipper:**

* **Merge Sort:**

**A screenshot of a computer program

Description automatically generated**

****

**A computer screen shot of a number

Description automatically generated**

* **Quick Sort:**

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. **Conclusion:**