**Project Plan Document**

**1. Executive Summary:**

The Tag Sort Algorithm project aims to develop a sorting technique where each element in an array is associated with a tag (or index), and sorting is performed based on the values of the elements. The project will involve implementing, testing, and documenting the algorithm. The expected outcome is a working and efficient algorithm capable of sorting any type of numerical or comparable data.

**2. Project Objectives:**

* Develop a functional implementation of the Tag Sort Algorithm in a programming language Python.
* Ensure the algorithm runs efficiently with a time complexity of O (n log n).
* Provide a clear flowchart, pseudocode, and modular breakdown for future maintainability.
* Test the algorithm using various datasets to validate correctness and performance.
* Document the process and deliverables.

**3. Project Scope:**

**3.1. In Scope:**

* Designing and developing the Tag Sort Algorithm.
* Implementing the algorithm in a programming language.
* Analyzing and optimizing the time and space complexity.
* Testing and validating the algorithm with multiple data sets.

**3.2. Out of Scope:**

* Implementation of parallel or distributed sorting techniques.
* Handling non-numerical data (unless it can be compared).
* Any advanced memory management optimizations (beyond O(n) space complexity).

**4.Milestones and Deliverables:**

| **Milestone** | **Deliverable** | **Planned Completion Date** |
| --- | --- | --- |
| **Milestone1:** Project Setup | Define scope and objectives, create initial setup. | August 19, 2024 |
| **Milestone2:** Algorithm Design | Develop flowchart, pseudocode, and system architecture. | August 26, 2024 |
| **Milestone3:** Implementation | Code the algorithm in the chosen language. | September 2, 2024 |
| **Milestone 4:** Testing | Test the algorithm with different datasets. | September 7, 2024 |
| **Milestone5:** Documentation | Finalize documentation, including SDD, and user guide. | September 16, 2024 |
| **Milestone6:** Final Review | Review the final implementation and make improvements. | September 28, 2024 |

**5. Project Schedule:**

**5.1. Task Breakdown:**

| **Task Name** | **Start Date** | **End Date** | **Duration** |
| --- | --- | --- | --- |
| Project Planning and Setup | August 13, 2024 | August 19, 2024 | 6 days |
| Design Algorithm Flow and Pseudocode | August 20, 2024 | August 26, 2024 | 5 days |
| Code Implementation | August 27, 2024 | September 2, 2024 | 6 days |
| Unit Testing & Validation | September 3, 2024 | September 7, 2024 | 4 days |
| Performance Optimization | September 8, 2024 | September11, 2024 | 4 days |
| Documentation and Review | September 12, 2024 | September 16, 2024 | 4 days |
| Final Delivery | September 28, 2024 | September 28, 2024 | 1 day |

**5.2. Gantt Chart:**

| **Task** | **Week 1** | **Week 2** | **Week 3** |
| --- | --- | --- | --- |
| Planning and Setup | ██████████ |  |  |
| Algorithm Design | ██████ |  |  |
| Code Implementation |  | ██████████ |  |
| Testing and Validation |  | ████ |  |
| Documentation |  |  | ██████ |
| Final Delivery |  |  | █ |
|  | | | |

**6. Resources Required:**

**6.1. Hardware/Software Resources**

* **Development Environment**: A computer system with IDE or text editors for coding.
* **Programming Language**: Python.
* **Version Control**: GitHub or similar for version control and collaboration.
* **Testing Framework**: Unit testing tools PyTest
* **Documentation Tools**: Microsoft Word
* **7. Quality Assurance**
* **Code Reviews**: Perform peer code reviews at each stage of development to ensure that the implementation meets the desired standards.
* **Automated Testing**: Use unit tests to validate the sorting functionality and ensure no regression in future changes.
* **Performance Benchmarking**: Test the algorithm with large datasets to check its performance, ensuring that it meets the O(n log n) time complexity.

**7. Communication Plan:**

* **Weekly Check-ins**: Regular meetings with stakeholders to review progress.
* **Documentation**: Provide detailed updates via project documents, including design, implementation, and testing.
* **Version Control**: Use a version control system to track changes and collaborate effectively.

**8. Final Deliverables:**

1. **Working Code Implementation**: A functional implementation of the Tag Sort Algorithm.
2. **System Design Document (SDD)**: A detailed document outlining the system architecture, design, and pseudocode.
3. **Testing and Validation Report**: A report of all tests performed, including test cases and results.
4. **User Guide**: A simple guide explaining how to use the algorithm.

**9. Conclusion:**

The Tag Sort Algorithm project will follow a structured approach with well-defined milestones and deliverables to ensure that the system is designed, implemented, and tested effectively. With the outlined resources and schedule, the project is expected to be completed by September 28, 2024.