



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

HUM -5151 Research Methodology and Technical Communication

(Technical Communication Component)

### **Synopsis (Review/Research Article)**

**Title** : Parallelization of Pigeonhole Sort for Efficient Data Sorting  
**Student Name** : Pasupuleti Rohith Sai Datta  
**Reg. No** : 230913003  
**Specialization** : CSE  
**Base Paper Title** : A New Innovation  
**Research Area** : Pigeonhole Sorting  
**Date of submission** : 06/09/2023  
**Mentor Details** : Prof. N. Gopalakrishna Kini

### **Abstract**

---

**Introduction:** Sorting algorithms play a crucial role in computer science and various applications. This research focuses on parallelizing the Pigeonhole Sort algorithm to enhance its efficiency in sorting large datasets. The motivation behind this work is the need for efficient sorting algorithms as data continues to grow exponentially.

**Motivation:** Traditional Pigeonhole Sort, while linear in time complexity, faces limitations with large datasets or wide value ranges. The motivation here is to leverage parallel computing techniques to make Pigeonhole Sort practical for large-scale data sorting.

#### **Methodology:**

**Data Distribution:** Input data is distributed among multiple processes in a distributed computing environment.

**Local Sorting:** Each process performs a local Pigeonhole Sort on its data segment.

**Global Min-Max Calculation:** Minimum and maximum values from each process are globally determined using MPI functions.

**Parallel Sorting:** Pigeonhole Sort is applied in parallel on each process, aligning pigeonhole ranges consistently.

**Merging Sorted Segments:** Sorted segments from each process are gathered and merged to form the final sorted dataset.

**Result:** Parallelization of Pigeonhole Sort showed significant improvements in sorting large datasets. This approach efficiently handles datasets with wide value ranges, making Pigeonhole Sort competitive in distributed computing.

**Conclusion:** Parallelized Pigeonhole Sort, implemented using MPI, offers a practical solution for efficiently sorting large datasets. It enhances the algorithm's capability to handle data with a considerable range of values, making it valuable for big data and distributed computing scenarios. This research contributes to the ongoing evolution of data processing techniques in modern computing.