**Module 3: Critical Thinking**

Chioma Chance

Colorado State University Global

CSC506: Design and Analysis of Algorithms

Dr. Banerjee

03/02/2025

**Introduction**

Efficiently managing patient records is crucial for healthcare systems, as quick access to sorted data impacts patient care. Sorting algorithms help structure records for optimal retrieval. This project compares Bubble Sort and Merge Sort in a hospital’s patient records system, analyzing their time complexity and performance to determine which is more suitable for handling medical data.

**Program Implementation**

I implemented both algorithms in Python, storing patient records as dictionaries within a list, sorted by either age or admission date. Bubble Sort iterates through the list multiple times, swapping adjacent elements if out of order, while Merge Sort follows a divide-and-conquer approach, breaking down and merging sorted sublists.

Bubble Sort’s simplicity made it easy to implement but inefficient for larger datasets due to its O(n²) time complexity. Merge Sort, though more complex, operates at O(n log n), making it significantly more efficient for large medical datasets.

**Challenges Faced**

Implementing Merge Sort correctly was challenging due to its recursive nature. Mistakes in merging led to incorrect outputs, emphasizing the importance of careful structuring. Additionally, structuring patient records for sorting required ensuring that the correct attributes were accessed. Reviewing pseudocode references helped integrate sorting while maintaining data integrity.

**Sorting in Healthcare Context**

Sorting is essential for managing patient records efficiently. Bubble Sort works best for small or nearly sorted datasets but is impractical for large hospital databases. Merge Sort, with its consistent O(n log n) runtime, is better suited for large-scale medical data management, where fast and stable sorting is critical in scenarios like emergency triage.

**Conclusion**

This project deepened my understanding of sorting algorithms and their real-world applications. Bubble Sort was easier to implement but inefficient for large datasets, while Merge Sort proved more efficient despite its complexity. Optimizing data structures enhances operational efficiency in critical environments like healthcare.

.

**References**

Lysecky, R., & Vahid, F. (2019, August). *Design and analysis of algorithms*. In R. Lysecky, & F. Vahid, *Data structures essential: Pseudocode with Python examples.* Zybooks. ISBN: 9781394012268

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT Press.

Patwary, M. A., Blair, J. R. S., Manne, F., & Gebremedhin, A. H. (2015). *Efficient sorting algorithms for large-scale data processing.* Journal of Parallel and Distributed Computing, 79, 1-14. <https://doi.org/10.1016/j.jpdc.2015.02.002>

A screen shot of a computer program

AI-generated content may be incorrect.Bubble Sort:

A screenshot of a computer program

AI-generated content may be incorrect.Merge Sort: