**Exploring Data Structures – Proof of Concept**

When evaluating sorting algorithms for a large dataset, such as the 1,000,000 product entries in a supermarket system, it’s important to consider both best-case and worst-case time complexities. **Merge Sort** stands out as the most efficient option due to its guaranteed **O(n log n)** performance in both best and worst cases. This is crucial when dealing with large datasets, as it ensures consistent and predictable sorting times, regardless of the initial arrangement of data.

In contrast, **Bubble Sort**, **Insertion Sort**, and **Selection Sort** all have worst-case complexities of **O(n²)**, meaning their performance degrades significantly with larger datasets. Although **Insertion Sort** can achieve **O(n)** in the best case (when the data is nearly sorted), this is a rare scenario in practice. **Bubble Sort** and **Selection Sort**, however, always perform at **O(n²)** in the worst case, making them highly inefficient for sorting large amounts of data.

Therefore, for a system like the supermarket, where sorting operations on large datasets might occur frequently, **Merge Sort** is the most practical choice. It provides consistent efficiency, handles larger datasets more gracefully, and ensures that the system’s performance remains robust even when sorting millions of product records.