**Sorting Algorithm Performance Report**

This report evaluates the performance of six sorting algorithms: BubbleSort, InsertionSort, MergeSort, IterativeMergeSort, QuickSort, and ShellSort. The primary goal is to compare their efficiency across varying input sizes and understand how their time complexity affects their performance.

**Algorithms and Their Complexity**

The following algorithms were tested, and their theoretical time complexities are listed:

* **BubbleSort**: O(n^2)
* **InsertionSort**: O(n^2)
* **MergeSort**: O(n log n)
* **IterativeMergeSort**: O(n log n)
* **QuickSort**: O(n log n) at average case, O(n^2) at worst case
* **ShellSort**: O(n^2)

**Experimental Setup**

* **Environment**:
  + OS: Linux-based
* **Input Sizes**: Vectors of size 10, 100, 1000, 5000, 10000, and 25000 were tested.
* **Metrics**: Execution time (in microseconds) was recorded for each sorting algorithm and input size.

**Results**

The results of the experiments are summarized in the table below:

텍스트, 라인, 도표, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

**Graphical Analysis**

A graph was generated to visualize the execution times of each algorithm across varying input sizes. It shows:

* **BubbleSort and InsertionSort**: These algorithms exhibit exponential growth in runtime as input size increases, with BubbleSort being the slowest.
* **MergeSort and IterativeMergeSort**: Both exhibit similar performance, with IterativeMergeSort being slightly faster due to reduced overhead.
* **QuickSort**: Demonstrates excellent average-case performance and is one of the fastest.
* **ShellSort**: Performs well but slightly lags behind QuickSort and MergeSort for larger inputs.

**Observations**

1. Algorithms with complexity (QuickSort, MergeSort, IterativeMergeSort) outperformed those with complexity, especially for larger input sizes.
2. Among algorithms, QuickSort exhibited the best performance for most input sizes.
3. BubbleSort consistently performed the worst, validating the inefficiency of algorithms for larger datasets.

**Conclusion**

This experiment demonstrates the practical impact of algorithm complexity on performance. QuickSort and MergeSort variants are optimal choices for large datasets, while BubbleSort and InsertionSort should be avoided for such cases.