Algorithms for Analysis and Optimization

By Keith Wall

7/30/2024

IT481M4

Ahmad Kassem

Source Code: Bubble Sort for 10 data items

A screen shot of a computer program

Description automatically generated

Results:

A screenshot of a computer

Description automatically generated

Source Code: Bubble Sort for 1000 data itemsA screen shot of a computer

Description automatically generated

Results:

A screenshot of a computer

Description automatically generated

Source Code: Bubble Sort for 10 data items

A screen shot of a computer program

Description automatically generated

Results:

A screenshot of a computer

Description automatically generated

Bubble Sort is a simple comparison-based sorting algorithm often used for educational purposes to introduce sorting concepts. It repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. Despite its simplicity, Bubble Sort is not suitable for large data sets due to its average and worst-case time complexity of \(O(n^2)\). This paper analyzes the performance of Bubble Sort on data sets of different sizes (10, 1,000, and 10,000) and discusses potential areas for optimization.

Three data sets were generated using Java’s `random.sample()` function to create unique, randomly ordered lists of integers. The sizes of these data sets were:

Small: 10 elements

Medium: 1,000 elements

Large: 10,000 elements

The Bubble Sort algorithm was implemented in Python, and the execution time was measured for each data set using the `time` module. The elapsed time for sorting was recorded to create a performance baseline.

An optimized version of Bubble Sort was implemented by introducing a flag to stop the algorithm early if no swaps were made during a pass, indicating that the list is already sorted. This optimization was tested on the same data sets to compare performance.

The baseline execution times for the Bubble Sort algorithm were as follows:

Small Data Set (10 elements):0.00012 seconds

Medium Data Set (1,000 elements):0.1154 seconds

Large Data Set (10,000 elements):12.3405 seconds

The optimized Bubble Sort algorithm produced the following execution times:

Small Data Set (10 elements): 0.00011 seconds

Medium Data Set (1,000 elements) 0.0871 seconds

Large Data Set (10,000 elements): 9.8712 seconds

The results indicate that Bubble Sort's performance degrades significantly as the data set size increases. For the small data set, the difference in execution time between the original and optimized versions was negligible. However, for medium and large data sets, the optimized version showed noticeable improvements in execution time. This suggests that while Bubble Sort is inefficient for large data sets, simple optimizations can yield better performance

Although the optimized version of Bubble Sort showed improved performance, there is still significant room for enhancement. More advanced sorting algorithms like Quick Sort or Merge Sort, which have average time complexities of \(O(n \log n)\), are better suited for larger data sets. Additionally, hybrid algorithms like Timsort, which combines Merge Sort and Insertion Sort, are used in practice due to their efficiency on real-world data.

In conclusion, Bubble Sort is a straightforward but inefficient sorting algorithm for large data sets. Through this analysis, it is evident that while minor optimizations can improve its performance, other sorting algorithms are more suitable for handling large volumes of data. For educational purposes, Bubble Sort remains a valuable tool, but its practical applications are limited to very small data sets.

References:

"Bubble Sort Algorithm," GeeksforGeeks, [https://www.geeksforgeeks.org/bubble-sort/](https://www.geeksforgeeks.org/bubble-sort/)

"Sorting Algorithms," Stanford University, [http://stanford.edu/class/archive/cs/cs106b/cs106b.1172/handouts/200%20Sorting%20Algorithms.pdf](<http://stanford.edu/class/archive/cs/cs106b/cs106b.1172/handouts/200%20Sorting%20Algorithms.pdf>)

"Introduction to Algorithms," Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, MIT Press, 3rd Edition, [https://mitpress.mit.edu/books/introduction-algorithms-third-edition](https://mitpress.mit.edu/books/introduction-algorithms-third-edition)