Common Sorting Algorithms

Michael Hinton, Cleveland State University
Jacob Katzenmeyer, Cleveland State University

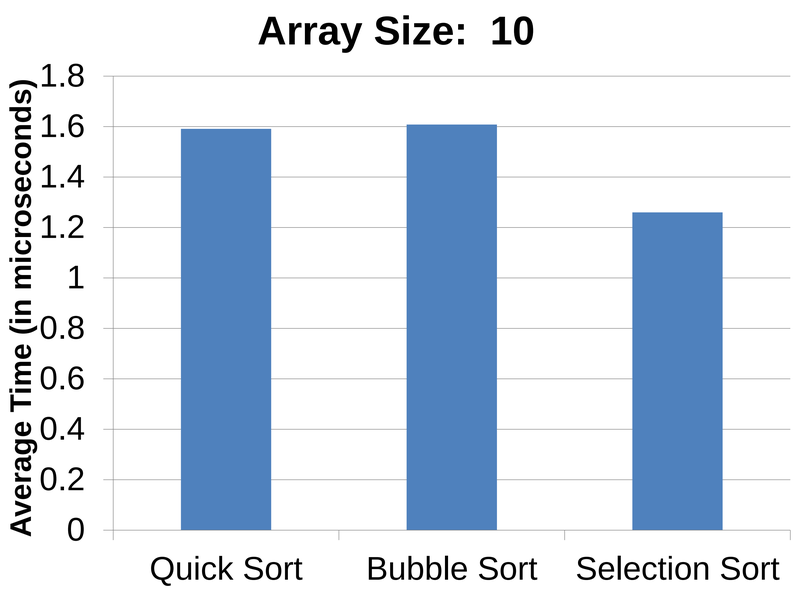
1. Introduction

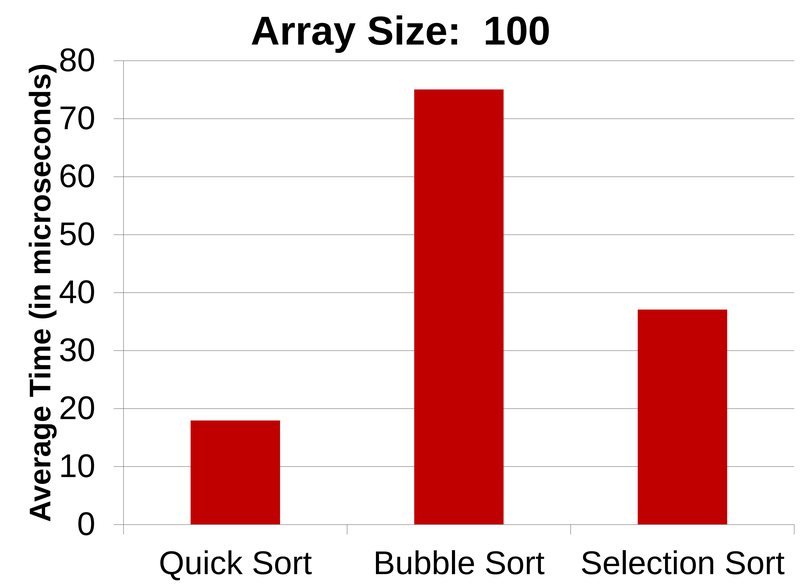
- Three common (and easy to implement) sorting algorithms are: Quick Sort, Bubble Sort, and Selection Sort.
- Average time complexities:
  - Quick Sort: O(n log n)
  - Bubble Sort: O(n^2)
  - Selection Sort: O(n^2)
- Big-O notation: Upper bound growth rate of a function.
- Quick Sort: Divide-and-conquer; recursively sort left and right sublists.
- Bubble Sort: Compares adjacent values and swaps them if necessary.
- Selection Sort: Divides list into two sublists: sorted and unsorted. Smallest value of the unsorted sublist is added to the end of the sorted sublist.

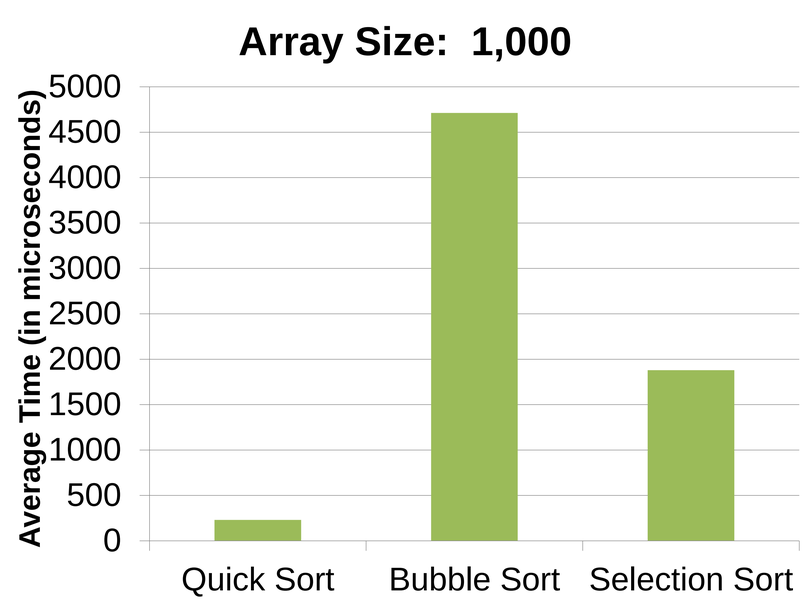
2. Methods

- Each algorithm sorts identical, randomly created arrays.
- The size of the array to be sorted is increased exponentially.
  - Sizes tested: 10; 100; 1,000
- Each size of array is tested 10,000 times and the quickest algorithm is recorded.
- The average time is also recorded.

- Run on a Dell Inspiron 15R







3. Results

10 Item Array

Average times:

- Quick Sort: 1.591 ms
  - Quickest 954 times
- Bubble Sort: 1.608 ms
  - Quickest 905 times
- **Selection Sort: 1.260 ms**
  - Quickest 6653 times

100 Item Array

Average times:

- **Quick Sort: 17.946 ms**
  - Quickest 9973 times
- Bubble Sort: 75.054 ms
  - Quickest 1 time
- Selection Sort: 37.081 ms
  - Quickest 26 times

1,000 Item Array

Average times:

- **Quick Sort: 230.256 ms**
  - Quickest 9997 times
- Bubble Sort: 4711.951 ms
  - Quickest 0 times
- Selection Sort: 1879.145 ms
  - Quickest 3 times

4. Conclusion

- Quick Sort was the fastest algorithm with larger data sizes.
- However, it was not the quickest algorithm at sorting a small data size (10).
- As expected (from their accepted time complexities), all algorithms are more than 10 times slower as the data size is increased 10 times.

Acknowledgments: Thanks to Choose Ohio First and Cleveland State University for the opportunity to conduct this research.