

Shelly Masih

CSCI 335 Report:

Algorithm	Input Size: 1k	Input Size: 31k	Input Size: 1M	QuickSelect Worst-Case
Half Selection Sort	0ms	2.5ms	150ms	-
Standard Sort	0ms	20ms	1000ms	-
Merge Sort	3ms	100ms	2000ms	-
In-Place Merge Sort	5ms	120ms	3000ms	-
Half Heap Sort	2ms	50ms	4000ms	-
Quick Select	1ms	50ms	2000ms	-
Worst-Case Quick Select	-	-	-	150ms

Algorithmic Analysis

1. Half Selection Sort

Algorithmic Analysis:

- Time Complexity: $O(n^2)$ in the worst case. This is because any input size for half selection sort will be proportional to the square of the input size.
- Performs $n(n-1)/2$ comparisons in the worst case.

2. Standard Sort

Algorithmic Analysis:

- Time Complexity: $O(n \log n)$ for average case. This is because after selecting a pivot, the array is split into 2 smaller sub arrays and during each call, an element is compared to the pivot.

3. Merge Sort

Algorithmic Analysis:

- Time Complexity: $O(n \log n)$ in all cases. This is because the sorting method requires the array to be split and then merged together after sorting.

4. In-Place Merge Sort

Algorithmic Analysis:

- Time Complexity: $O(n \log n)$ in all cases. In-place merge saves time because it stops once the median is found compared to doing the whole sort.
- In-place merge doesn't require extra space.

5. Half Heap Sort

Algorithmic Analysis:

- Time Complexity: $O(n \log n)$ in all cases. This is because after building a heap, removes $n/2$ elements from the heap and this is performed repeatedly.

6. QuickSelect

Algorithmic Analysis:

Time Complexity:

- $O(n^2)$ in the worst case. This is due to picking a poor pivot
- $O(n)$ average case. This is due to picking a pivot using the median of 3.

7. Worst-Case QuickSelect

Algorithmic Analysis:

- Generates a worst-case input for QuickSelect using median-of-3.

Comparison and Observations

- Merge Sort and In-Place Merge Sort performs well with $O(n \log n)$ time complexity.
- QuickSelect shows average-case performance but has a potential worst-case scenario.
- I noticed that the actual timed results were a lot shorter compared to what I expected.
This could be due to an error on my end or this might just be the way it is but this is what I picked up after testing.