Cross-Review Summary: Selection Sort vs Insertion Sort

Pair:

Student A: Insertion Sort (with early termination) — *Gaziza*

Student B: Selection Sort (with early termination) — *Sultan*

1. Algorithm Overview

Feature	Selection Sort (Sultan's Algorithm)	Insertion Sort (Gaziza's Algorithm)
Approach	Repeatedly selects the minimum element and places it at the beginning.	Inserts each element into the correct position in the sorted portion.
In-place	✓Yes	✓Yes
Stable	✗No	✓Yes
Adaptive	✗No	✓Yes
Time Complexity	Best: $\Omega(n)$ Average: $\Theta(n^2)$ Worst: $O(n^2)$	Best: $\Omega(n)$ Average: $\Theta(n^2)$ Worst: $O(n^2)$
Space Complexity	O(1)	O(1)
Key Strength	Fewer swaps (1 per pass)	Excellent on nearly sorted data
Key Weakness	Always performs $n^2/2$ comparisons	Many shifts when data is reverse-sorted

2. Empirical Comparison

Input Size (n)	Selection Sort (ms)	Insertion Sort (ms)	Faster Algorithm
100	0.7	1	Seletion Sort
1,000	4.7	10	Seletion Sort
10,000	85.4	61	Seletion Sort
50,000	514.8	1440	Seletion Sort

3. Theoretical Comparison

Metric	Selection Sort	Insertion Sort	Observation
Comparisons	~$n^2/2$ (constant regardless of order)	Variable; fewer if array is nearly sorted	Insertion Sort adapts better
Swaps/Shifts	$n - 1$ (one per iteration)	Up to $n^2/2$ shifts	Selection Sort reduces swap overhead
Best Case	$\Omega(n)$ (already sorted)	$\Omega(n)$ (already sorted)	Both can early terminate
Worst Case	$O(n^2)$	$O(n^2)$	Same asymptotic bound
Practical Behavior	Stable runtime; predictable	Faster on partially sorted data	Insertion Sort usually wins empirically

3. Joint Conclusion

Aspect	Summary
Correctness	Both algorithms produce identical sorted output.
Empirical Validation	Time and comparisons confirm theoretical O(n^2) growth.
Optimizations	Early termination and reduced overhead improve efficiency without changing complexity.
Algorithmic Trade-off	Selection Sort minimizes writes; Insertion Sort minimizes comparisons.
Best Use Case	Selection Sort: small datasets, minimal write environments. Insertion Sort: partially sorted or small datasets requiring stability.

Final Observation:

Insertion Sort provides better real-world performance, while Selection Sort is simpler and more predictable. Both are educationally valuable for illustrating algorithmic analysis and asymptotic behavior.