# Algorithmic Analysis and Peer Code Review – Pair 1

# Student A: Zhanabek Eldos

# Student B: Amanbek Nazarbek

# Course: Design and Analysis of Algorithms Assignment 2: Algorithmic Analysis and Peer Code Review

# Algorithm Analysis Report: Insertion Sort vs Selection Sort

## 1. Algorithm Overview

|  |  |  |
| --- | --- | --- |
| Aspect | Insertion Sort (Optimized) | Selection Sort (with Early Termination) |
| Category | Quadratic Sort (incremental insertion) | Quadratic Sort (selection-based) |
| Key Idea | Inserts each element in its correct place using binary search for position finding. | Selects the minimum element and places it at the beginning. |
| Optimization | Binary search reduces comparisons to O(n log n). | Early termination if no swaps occur. |
| In-place | Yes | Yes |
| Stable | Yes | No |

## 2. Theoretical Complexity Analysis

|  |  |  |
| --- | --- | --- |
| Case | Insertion Sort (Binary Search) | Selection Sort (Early Stop) |
| Best Case | Ω(n) – minimal shifts | Ω(n²) – full scan needed |
| Average Case | Θ(n²) with O(n log n) comparisons | Θ(n²) |
| Worst Case | O(n²) | O(n²) |
| Auxiliary Space | Θ(1) | Θ(1) |
| Swaps | Up to n²/2 shifts | n-1 swaps |
| Comparisons | O(n log n) | O(n²) |

## 3. Mathematical Derivation

For Insertion Sort (Optimized):  
Each insertion takes O(log i) comparisons and O(i) shifts.  
Total time: T(n) = Σ (log i + i) = O(n²)  
  
For Selection Sort:  
Each iteration requires n - i comparisons.  
Total time: T(n) = (n-1) + (n-2) + ... + 1 = O(n²)

## 4. Empirical Validation (Simulated Results)

|  |  |  |
| --- | --- | --- |
| n | InsertionSortOptimized (ms) | SelectionSort (ms) |
| 100 | 0.05 | 0.07 |
| 1,000 | 2.8 | 4.5 |
| 10,000 | 240 | 410 |
| 100,000 | 25,000 | 40,000 |

Trend: InsertionSortOptimized performs better, especially on nearly sorted arrays.

## 5. Code Review & Optimization

InsertionSortOptimized:  
Clean and readable code.  
Proper binary search.  
 Can add performance tracking and early termination check.  
  
SelectionSort:  
 Simple structure.  
 Add 'swapped' flag for early termination.  
Track metrics with PerformanceTracker.

## 6. Conceptual Performance Graph

Time (T)  
│  
│ SelectionSort  
│ /  
│ /  
│ /  
│ /  
│ / InsertionSortOptimized  
│\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ n

## 7. Conclusion

InsertionSortOptimized outperforms SelectionSort in best and average cases.  
SelectionSort remains simpler but less efficient even with early termination.  
  
Winner: InsertionSortOptimized.