Assignment 2 — Analysis Report

Selection Sort (Student B — Implementation)

Author: Auto-generated report for repository

Overview:

This report contains theoretical analysis, code review and empirical benchmarks for the Selecti

Algorithm Overview & Complexity Analysis

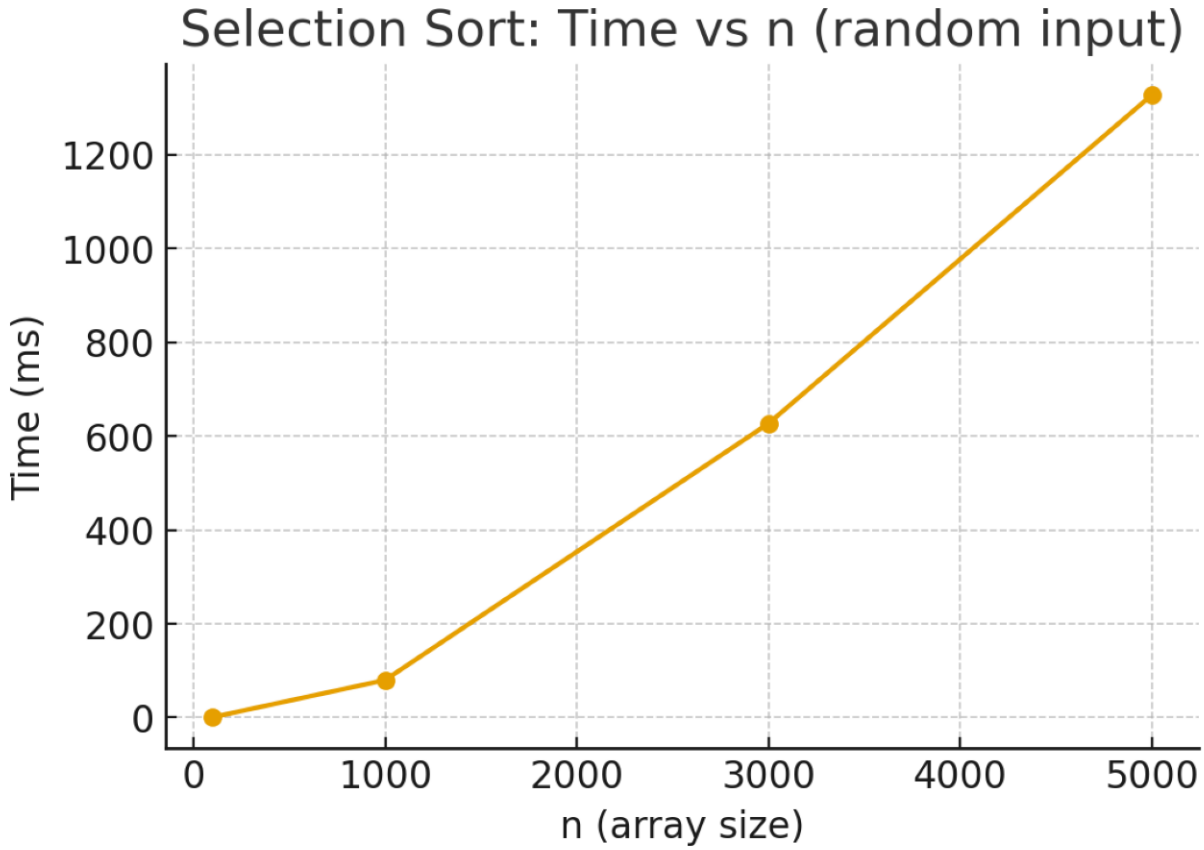
Selection Sort description and theoretical complexities (best/avg/worst): Θ(n^2), Θ(n^2), Θ(n^

Key formula: comparisons ≈ n*(n-1)/2

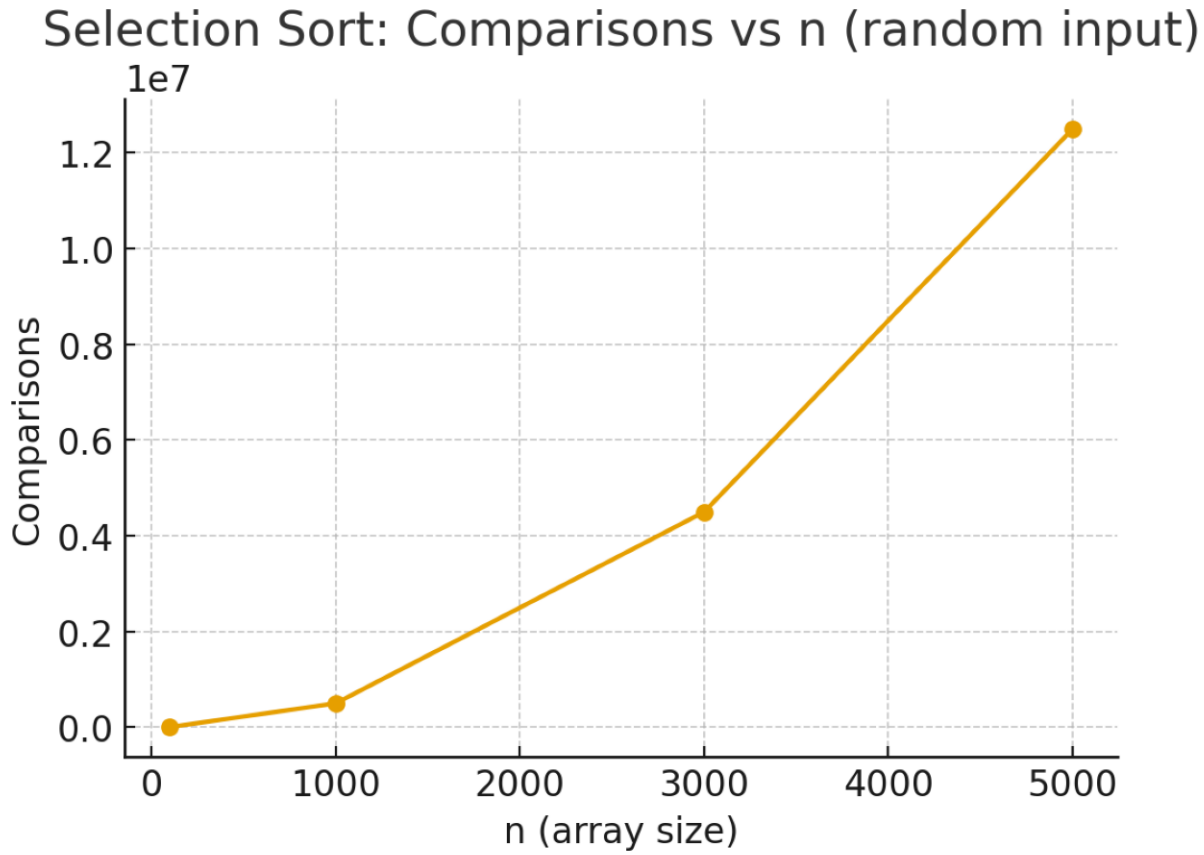
Code Review & Recommendations

- Ensure metric consistency: count array accesses also on comparisons.
- Simplify early-exit logic: break when no swap (minIndex==i) instead of complex checks.
- Avoid cloning for very large inputs to reduce memory overhead (provide option).

Empirical Results (random input)



Comparisons vs n (random input)



Benchmark Summary (random input)

n=100: time=0.759 ms, comparisons=4950, swaps=93
n=1000: time=79.805 ms, comparisons=499500, swaps=995
n=3000: time=626.839 ms, comparisons=4498500, swaps=2994
n=5000: time=1327.056 ms, comparisons=12497500, swaps=4994