Complexity and Mutation Analysis

Bubble Sort Complexity

Bubble sort has a time complexity of $O(n^2)$ in the worst and average cases, and $O(n)$ in the best case (when the list is already sorted).

Quick Sort Complexity

Quick sort has an average time complexity of $O(n \log n)$, but can degrade to $O(n^2)$ in the worst case (when the pivot selection is poor).

Worklist Algorithm Analysis

The worklist algorithm processes a matrix by continuously updating its values based on the sum of elements in the same row and column. The time complexity is $O(n^3)$ in the worst case.

Mutation Explanation

Bubble Search Mutant

File: bubble_search_mutant.py

Bug: Changed the comparison operator in bubble sort from > to <

Effect: Causes the list to be sorted in descending order instead of ascending order

Why it fails: The binary search function expects the input list to be sorted in ascending order. When given a descending list, the binary search will not work correctly.

Quick Search Mutant

File: quick_search_mutant.py

Bug: Changed the pivot selection to the last element but didn't adjust the list slicing

Effect: The pivot element is included in both left and right partitions, causing incorrect sorting

Why it fails: The quick sort implementation produces an incorrectly sorted list, which causes the binary search to fail.

Verification of Mutants

To verify that the mutants fail the tests:

1. For `bubble_search_mutant.py`:
    - Input: [5, 3, 8, 4, 2], target=4
    - Expected: Should return index 2 (after sorting: [2, 3, 4, 5, 8])
    - Actual: Sorts to [8, 5, 4, 3, 2], binary search fails to find 4
2. For `quick_search_mutant.py`:
    - Input: [9, 1, 6, 3], target=1
    - Expected: Should return index 0 (after sorting: [1, 3, 6, 9])
    - Actual: Incorrect sorting causes binary search to fail