

## Assignment 3 report - CMPT 276 - Duc Duy Pham

### Testing Algorithms: QuickSort and MergeSort

In assessing the effectiveness of sorting algorithms, two widely used algorithms were chosen for implementing and testing in this assignment: QuickSort and MergeSort, as they satisfy the requirement for  $O(\log n)$  run time.

#### I. Functional Testing

For both QuickSort and MergeSort, tests were designed to cover a variety of input conditions, including but not limited to, arrays containing negative numbers, mixed positive and negative numbers, duplicates, elements of the same value, large numbers, arrays in sorted and reverse order, and strings.

#### Coverage Results:

For only functional testings, the coverage results for the entire maven project was 85.7%, with 13 different tests for both MergeSort and QuickSort. This shows that functional testing covered a broad variety of functionality that both sorting algorithms should cover. However, the remaining 14.3% lies on the structure of the code itself, showing the need for structural testing.

#### II. Structural Testing

**QuickSort:** The structural tests, especially those focusing on the pivot as the smallest and largest elements, provided deeper coverage of the recursive partitioning logic. This ensured that edge cases, which could potentially reduce performance or correctness, were addressed.

**MergeSort:** Tests targeting the divide-and-conquer algorithm, particularly the merging of odd and even-length subarrays and nearly sorted arrays, expanded coverage to essential yet less obvious paths within the algorithm. This included verifying the merge process's ability to handle arrays of differing lengths and compositions.

#### Coverage results:

After adding the structural tests for Quicksort, the coverage for the entire project went up to 87.5%, showing that structural tests for MergeSort will fill up the rest of the test cases. After adding structural tests for MergeSort, the coverage went up to 99.1%, which shows that the tests covered almost every case in the project.