# Question 3 – Checklist Table

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tasks | Mark | Checklist  (Yes/no) and person(s) to work on this task |
| 1 | Quick-select algorithm for searching the *k*-th element.  - Correct implementation of searching *k*-th element using quick-select algorithm.  - Output the intermediate results for a random array of 10 elements for tutor to inspect the correctness of the algorithms. | 4 |  |
| 2 | Test the Quick-select algorithm with  - Different array sizes (10,000, 100,000, 1,000,000, etc.) that can show significant results.  - Different pivot (random pivot vs fixed pivot).  - Different cases (e.g. best, average, and worst). | 4 |  |
| 3 | Merge-sort algorithm for searching the *k*-th element.  - Correct implementation of searching *k*-th element using merge-sort algorithm.  - Output the intermediate results for a random array of 10 elements for tutor to inspect the correctness of algorithms. | 4 |  |
| 4 | Test the Merge-sort algorithm with  - Different array sizes (10,000, 100,000, 1,000,000, etc.) that can show significant results.  - Test the Merge-sort algorithm with different cases (e.g. best, average, and worst). | 4 |  |
| 5 | - Include the above experiment results that can be used to perform a comparative analysis (such as drawing the graphs for comparison) between the two algorithms (Quick-select & Merge-sort) in the report.  - Conclude your findings in the report. | 4 |  |

# Quick-select algorithm

## Array size test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Array size, *n* | 10,000 | 100,000 | 1,000,000 | 5,000,000 | 10,000,000 |
| Duration (s) |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pivot selection | Random | Random | Random | Random | Fixed |
| Pivot selected |  |  |  |  |  |
| Duration (s) |  |  |  |  |  |

## Pivot test (same set of array)

|  |  |  |  |
| --- | --- | --- | --- |
| Case | Best | Average | Worst |
| Duration (s) |  |  |  |

## Test of different cases (same set of array)

# Merge-sort algorithm

## Array size test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Array size, *n* | 10,000 | 100,000 | 1,000,000 | 5,000,000 | 10,000,000 |
| Duration (s) |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Case | Best | Average | Worst |
| Duration (s) |  |  |  |

## Test of different cases (same set of array)

# Comparative analysis

# Conclusion