Part 1 & 2

A screenshot of a computer

AI-generated content may be incorrect.

Part 3

Sort Type | Data Struct | Operation | Runtime (ms)

-----------------------------------------------

QUICK LINKEDLIST 71160 11

INSERTION LINKEDLIST 6219111 98

MERGE LINKEDLIST 55176 39

SELECTION LINKEDLIST 12502492 250

QUICK ARRAY 12502492 11

INSERTION ARRAY 6342526 98

MERGE ARRAY 55211 2

SELECTION ARRAY 12497500 23

Sort Type | Data Struct | Operation | Runtime (ms)

-----------------------------------------------

QUICK LINKEDLIST 72226 3

INSERTION LINKEDLIST 6298055 71

MERGE LINKEDLIST 55192 11

SELECTION LINKEDLIST 12502495 113

QUICK ARRAY 12502495 3

INSERTION ARRAY 24888484 71

MERGE ARRAY 120462 2

SELECTION ARRAY 49995000 52

Sort Type | Data Struct | Operation | Runtime (ms)

-----------------------------------------------

QUICK LINKEDLIST 74984 2

INSERTION LINKEDLIST 6171418 35

MERGE LINKEDLIST 55201 3

SELECTION LINKEDLIST 12502488 142

QUICK ARRAY 12502488 2

INSERTION ARRAY 625897199 35

MERGE ARRAY 718392 7

SELECTION ARRAY 1249975000 759

Sort Type | Data Struct | Operation | Runtime (ms)

-----------------------------------------------

QUICK LINKEDLIST 70060 2

INSERTION LINKEDLIST 6227651 56

MERGE LINKEDLIST 55240 6

SELECTION LINKEDLIST 12502492 115

QUICK ARRAY 12502492 2

INSERTION ARRAY 2498834288 56

MERGE ARRAY 1536693 21

SELECTION ARRAY 4999950000 3167

Process finished with exit code 0

Part 4

REPORT  
  
expected complexity:

* Quick and merge sort is logarithmic O(n log n) where n is the size of the input.
* Insertion and selection sort is quadratic O(n^2)

Comparison:

As expected, as the elements in the array or LinkedList increases, the runtime increases. Quick and merge sorts are ideal for larger data sets while insertion and selection sorts are ideal for smaller data sets because of their respective time complexity (Logarithmic vs Quadratic). Logarithmic complexity will be slower at the start but become increasingly efficient as the data set is iterated through. This lab made us aware of how important time complexity is when dealing with large data sets.