

**GTU Department of Computer Engineering**

**CSE 222/505 - Spring 2020**

**Homework 6 Report**

**Q2**

**Buğra Eren Yılmaz**

**1801042669**

# 1. Summary

The linked list data structure is better suited for merge sort, in contrast to normal array structure is better suited for quick sort.

The reason for this is that the linked list data structure is very bad at random access (unless you use some kind of iterator for accessing). Because of this, it is worse on quicksort but better on merge sort. Because merge sort has less random access than quicksort. But this is totally opposite for array structures, where the random access is not a problem. Because of this nature of array structures it is more suited to use quicksort on them.

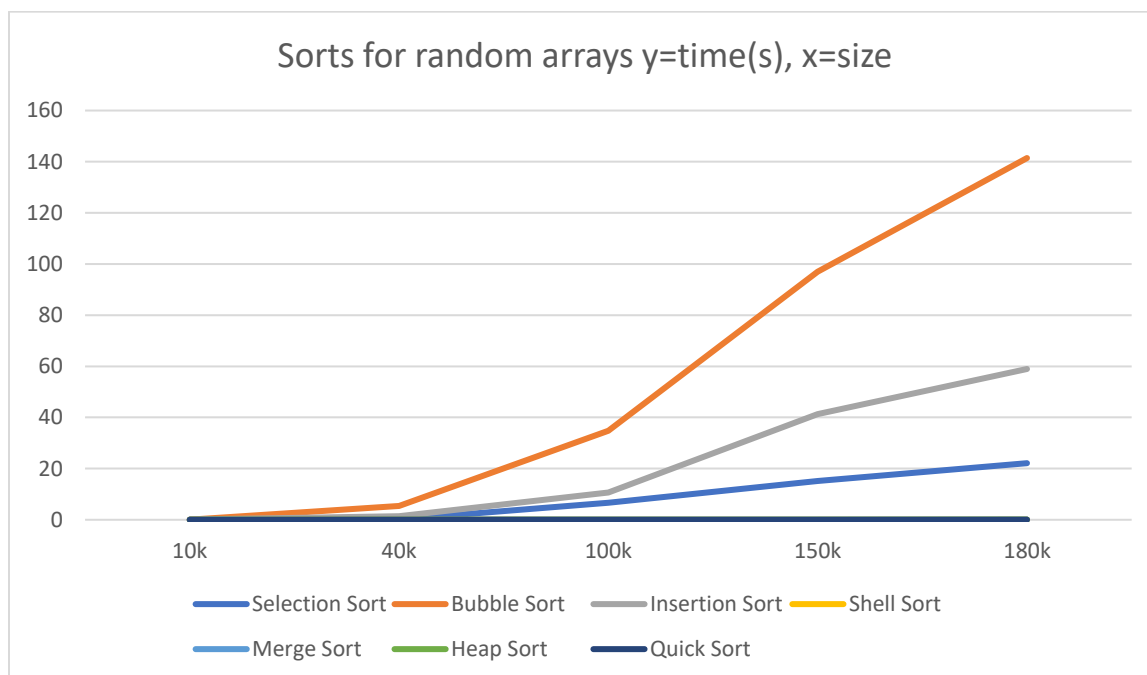
The benchmarks took ages, even on a machine with 6 core cpu @3.8ghz. I run the tests with max heap of 8gb of ram by giving the JVM -Xmx8G flag.

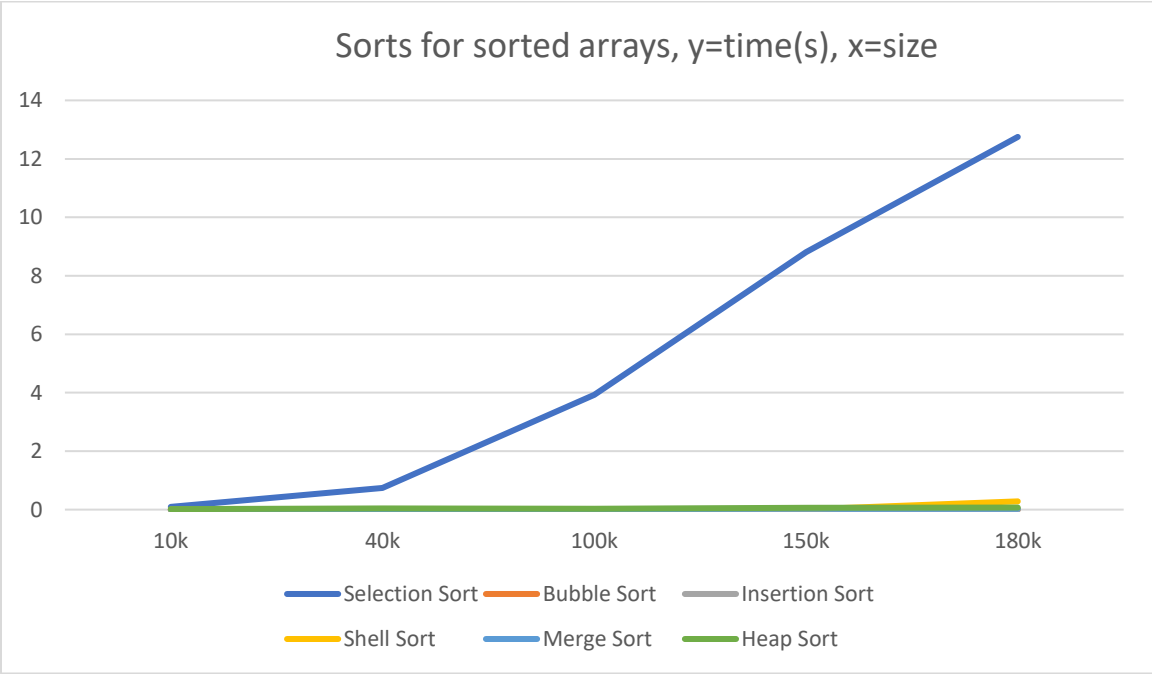
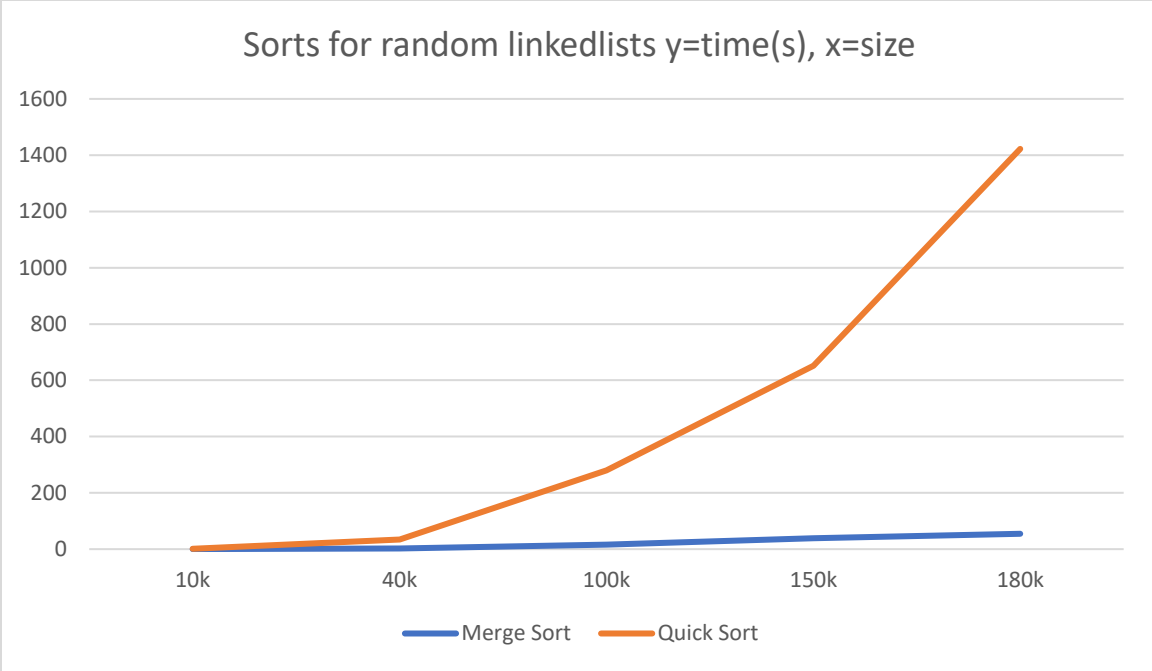
Even on that amount of ram the books quick sort function failed on sorted arrays of 40k size and above. Because it tried to partition 1 by 1, and stackoverflowed. My quicksort implementation for linked list is working fine.

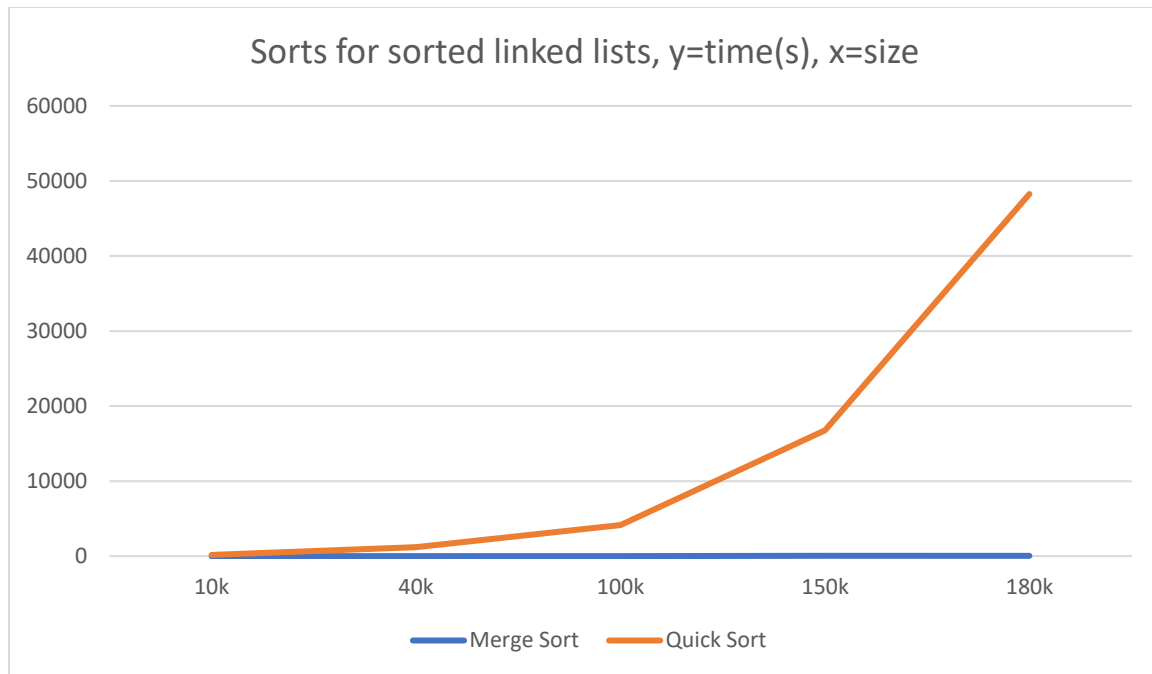
Graphs provided below. Output of program and the source of program is also provided in submitted files.

Graphs uses the average of 20 runs for every sort. You can also find the 20 runs result separately in provided output of program.

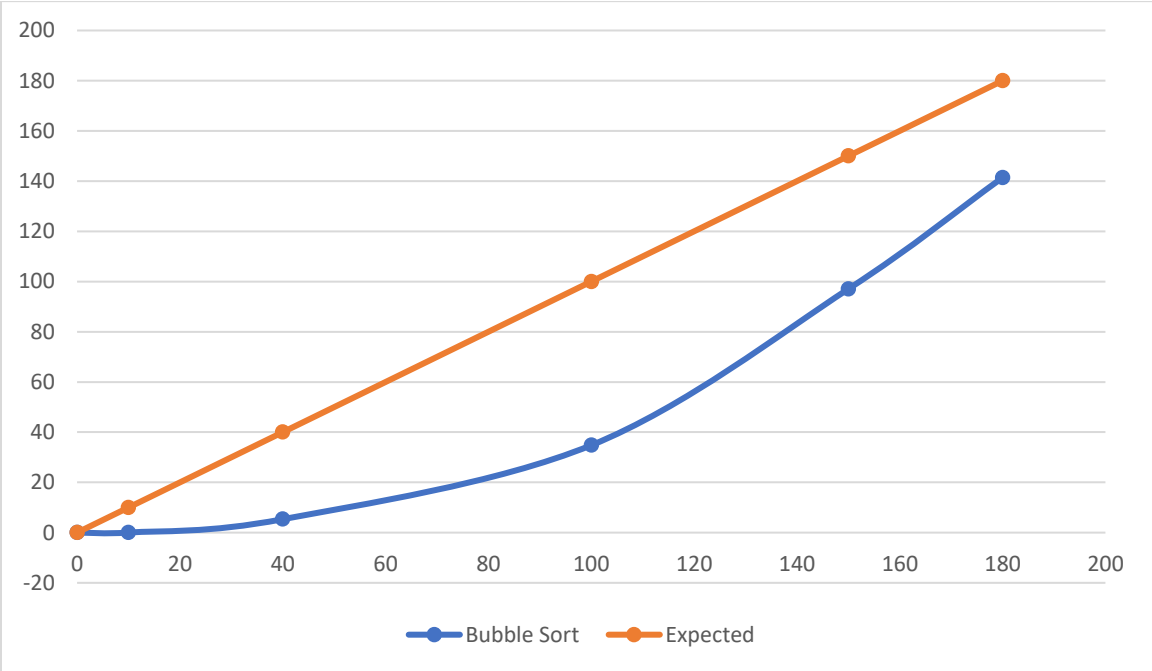
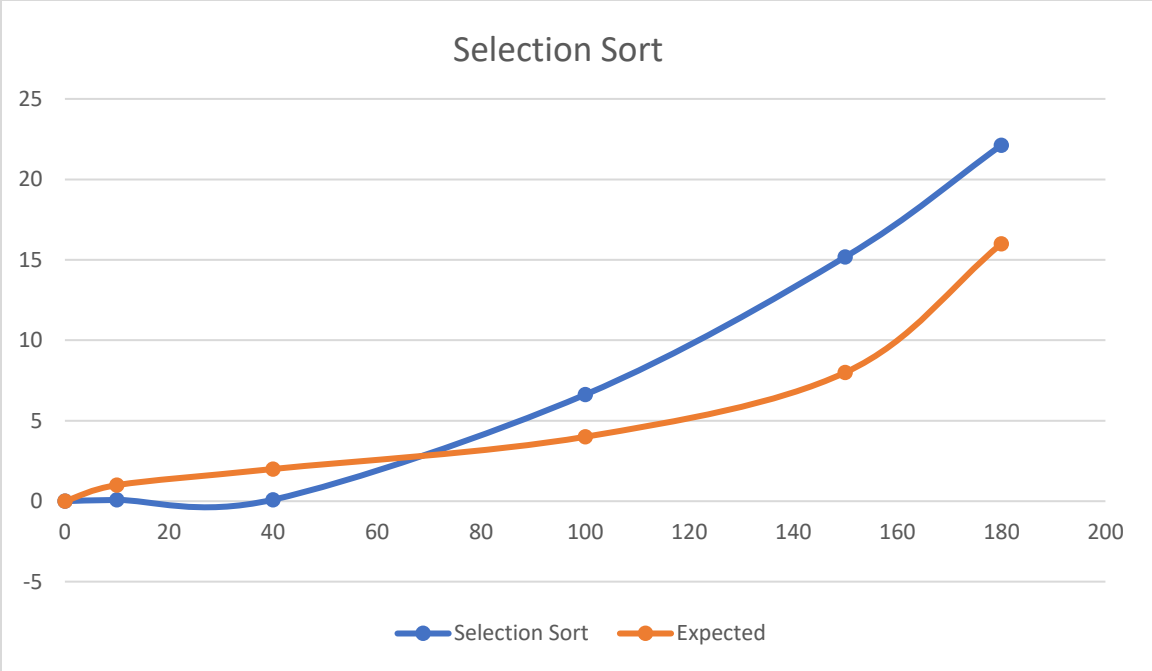
## 2. General graphs

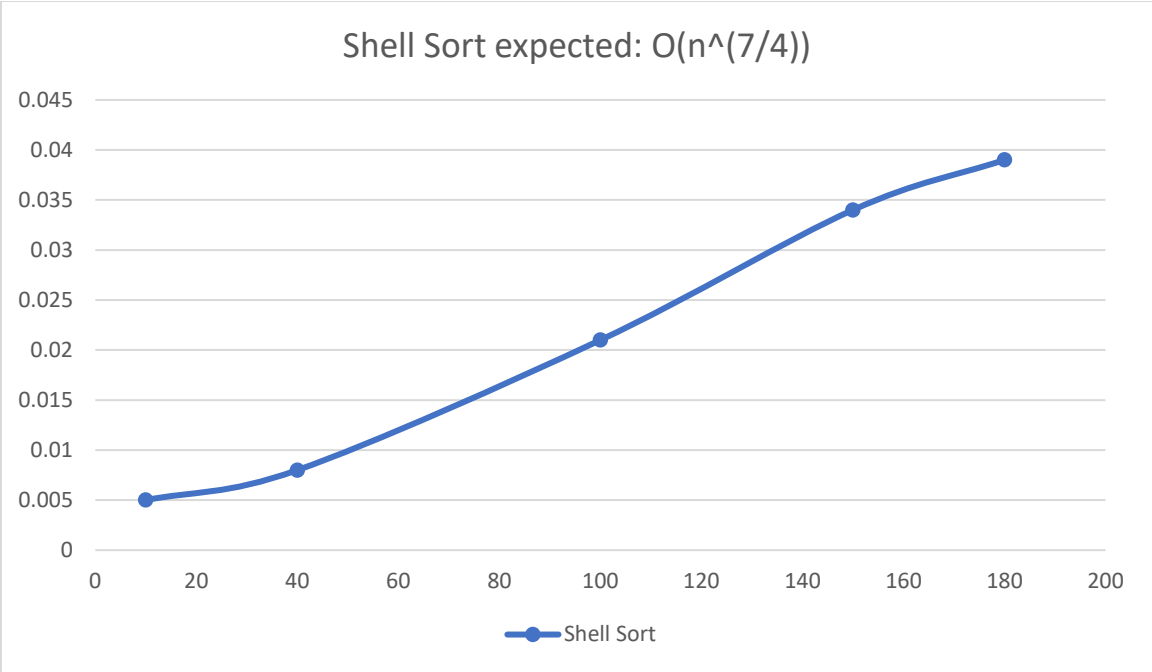
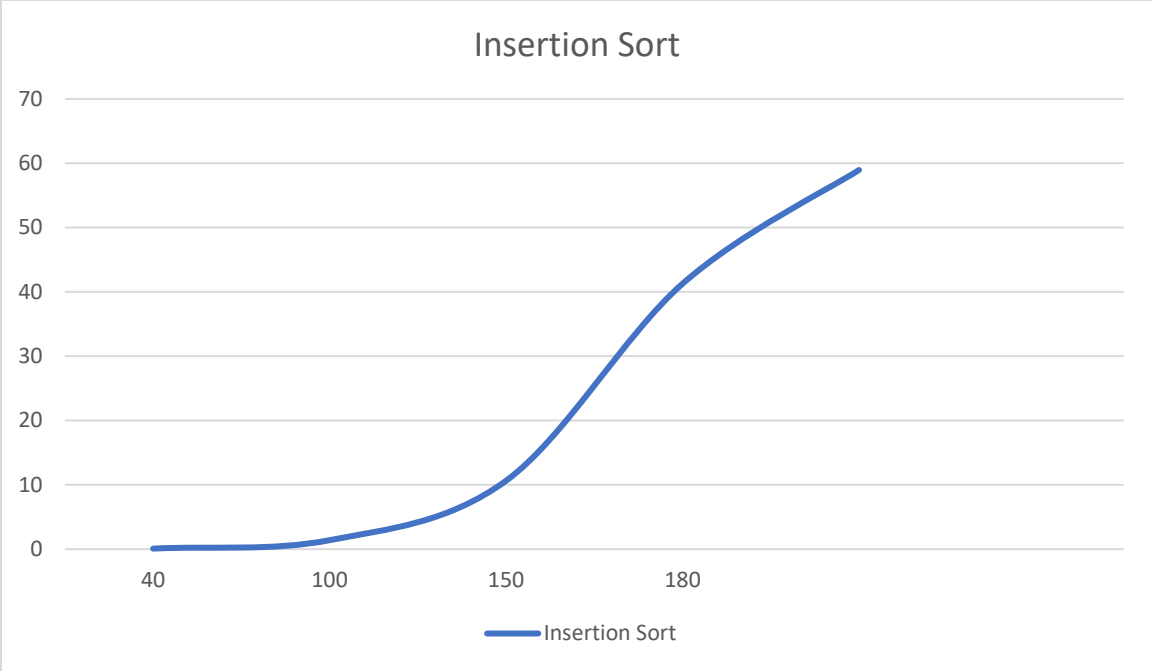


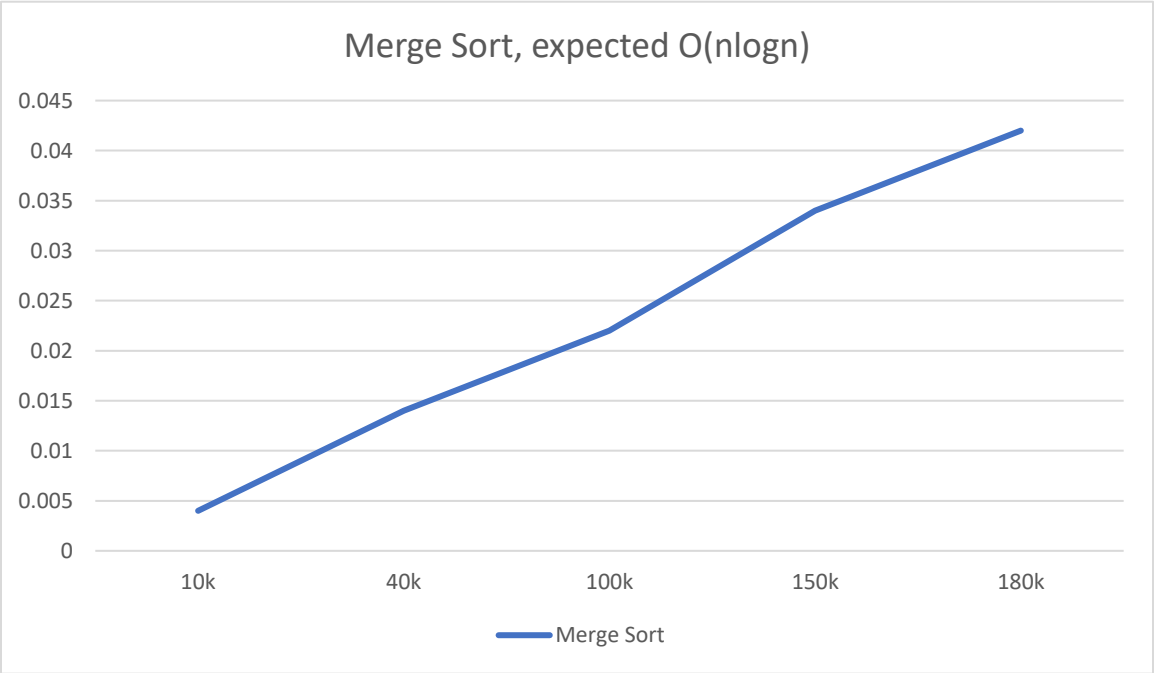
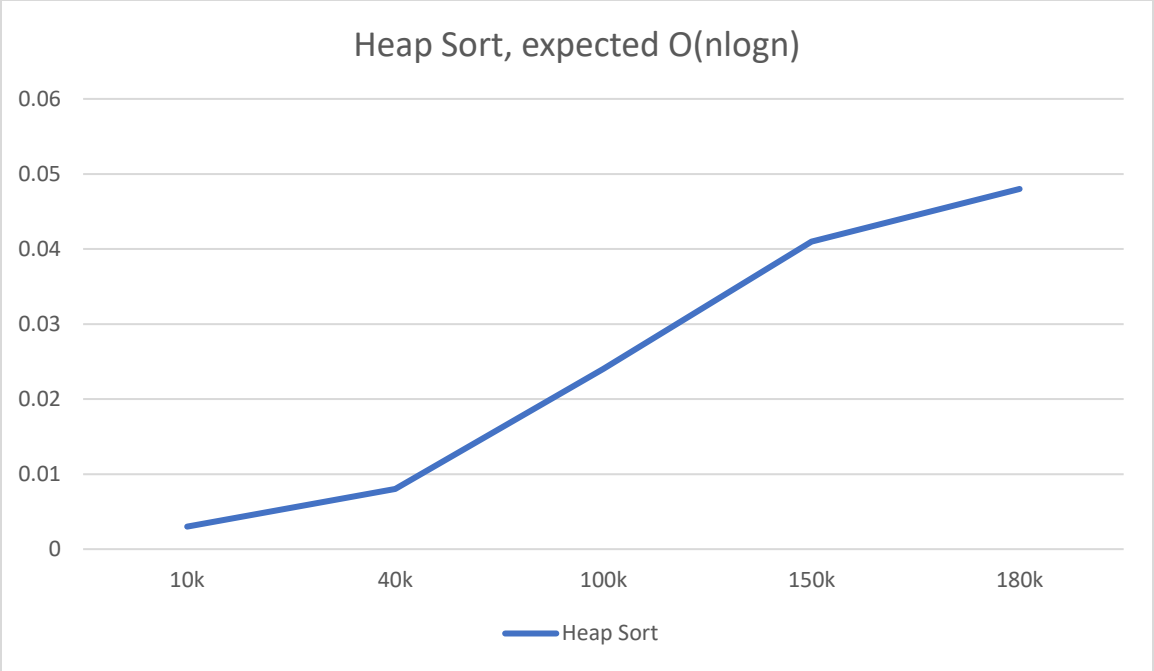




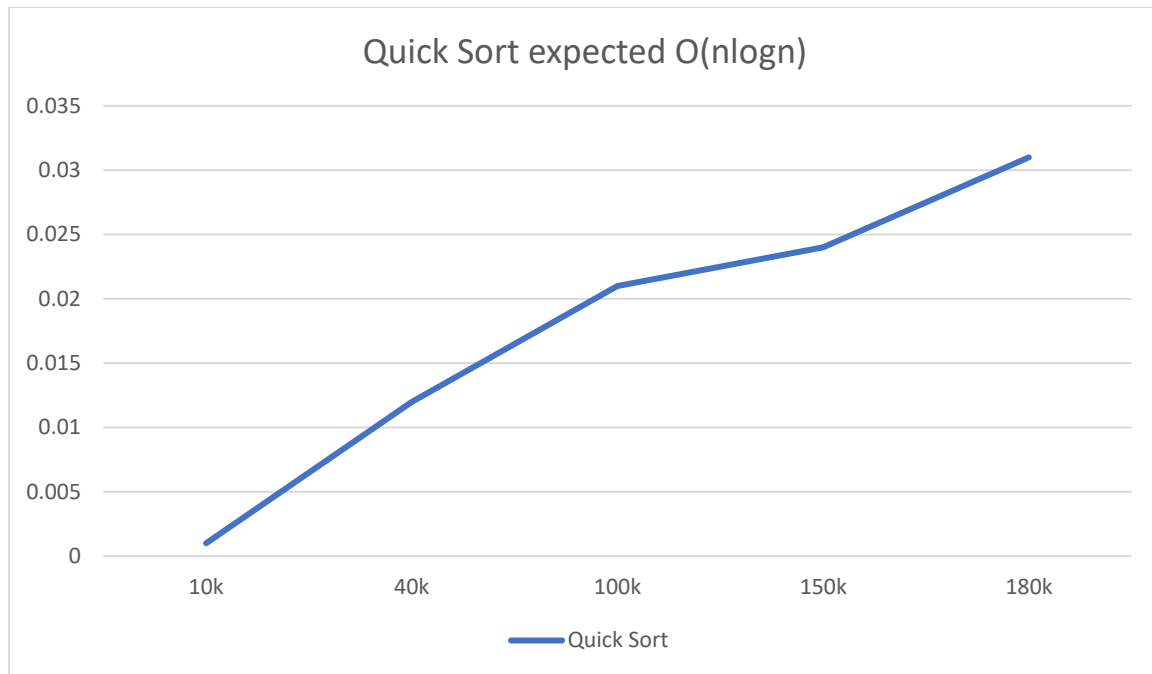
### 3. Individual Graphs for random arrays



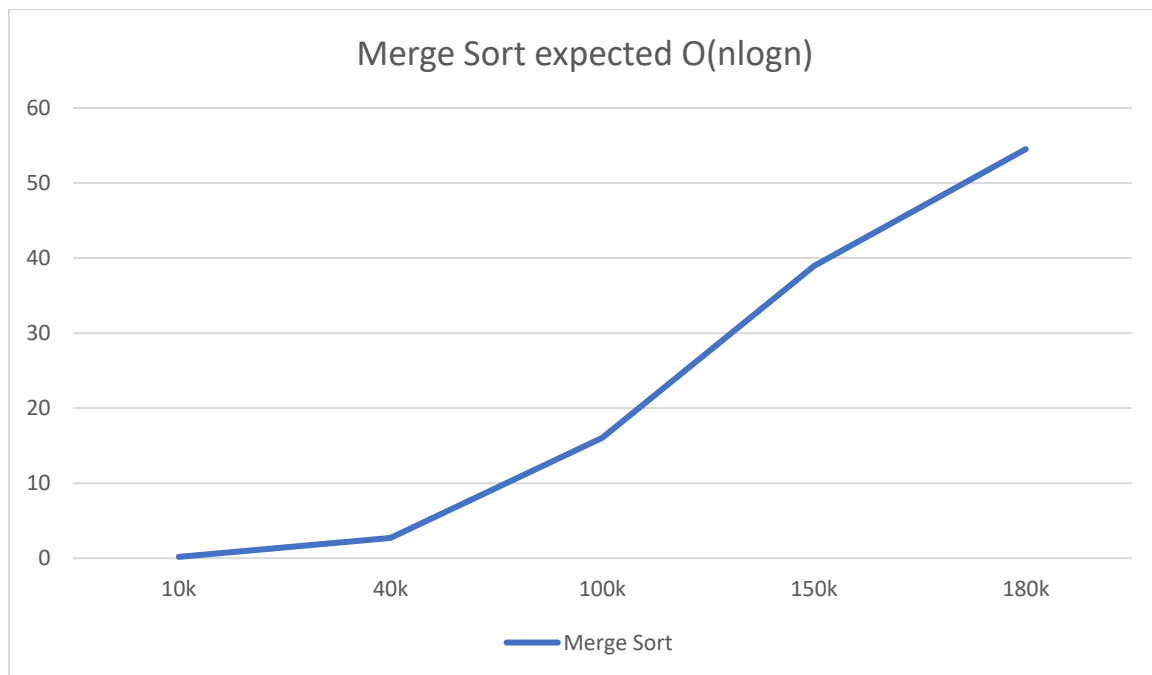


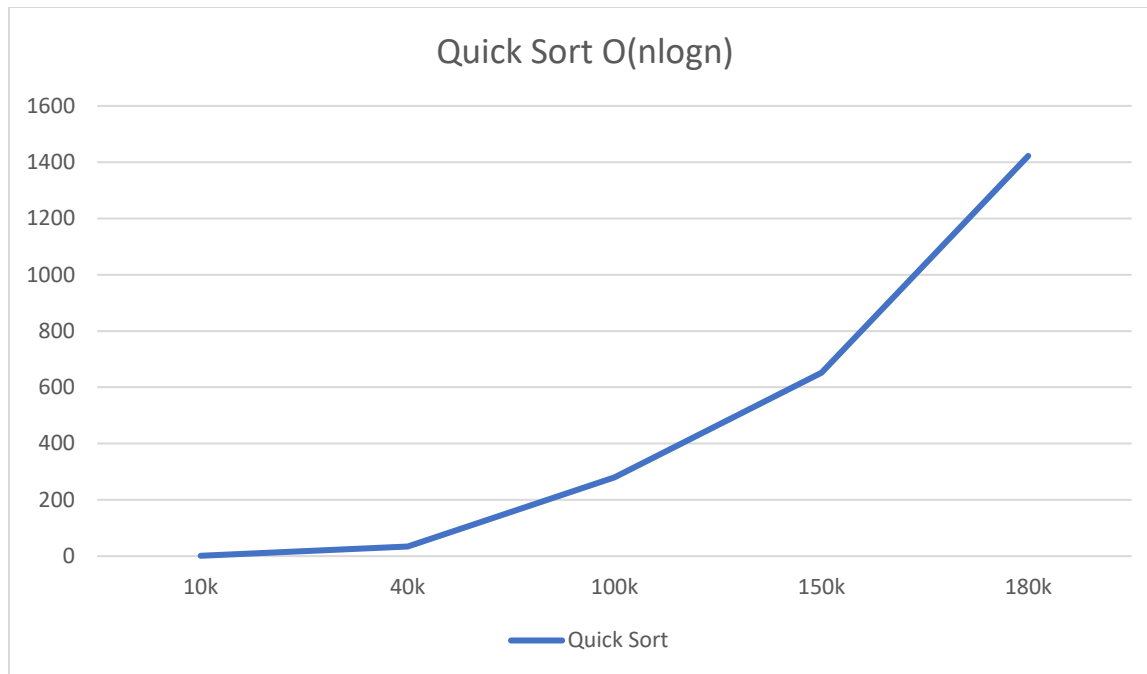




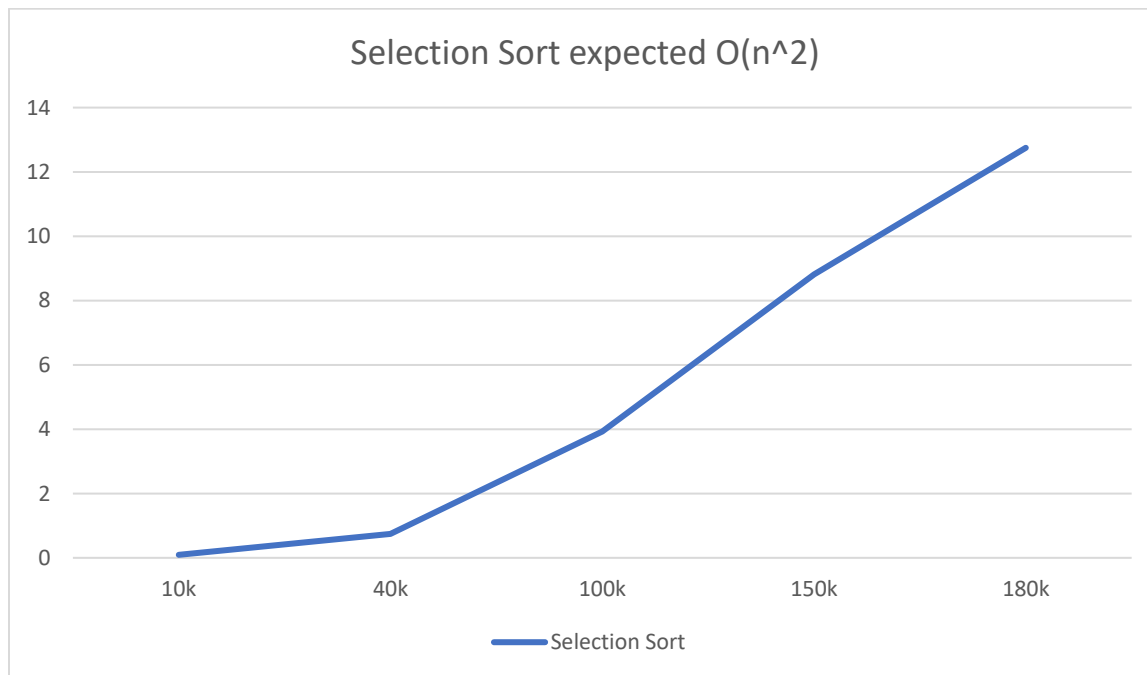


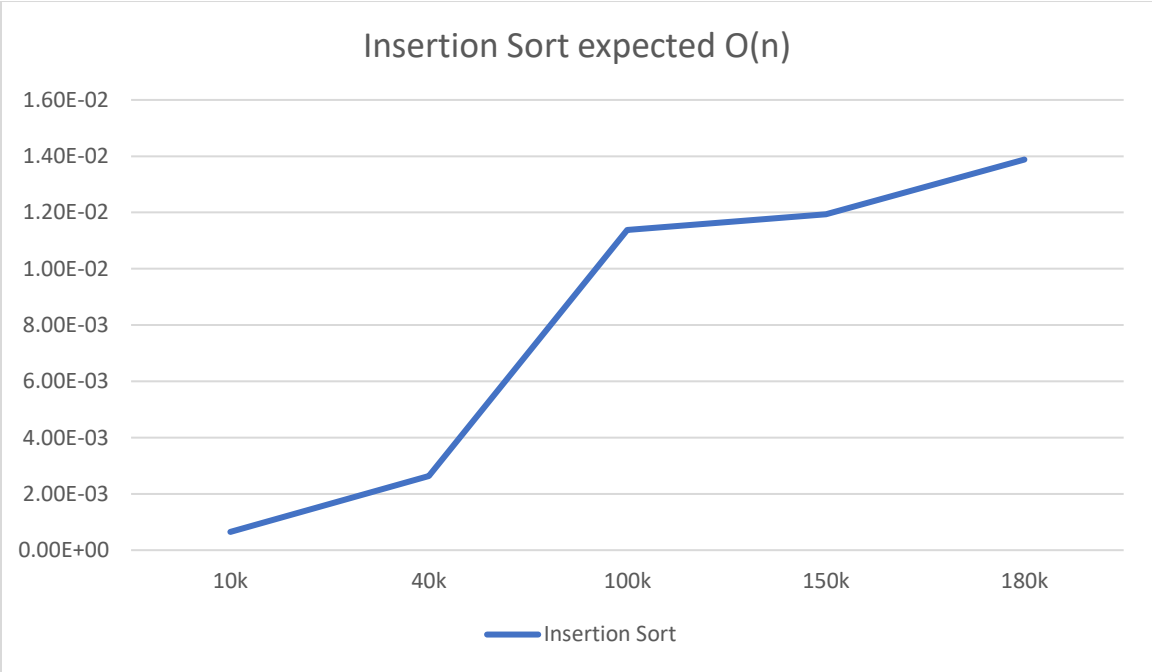
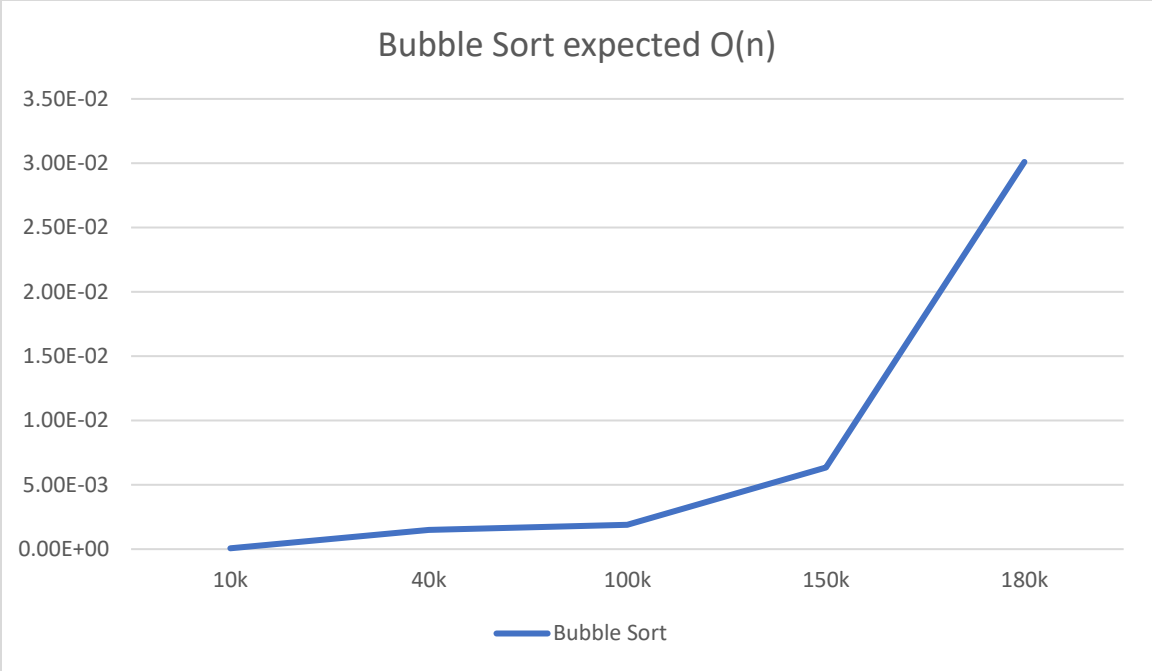
## 4. Individual graphs for random Linkedlists

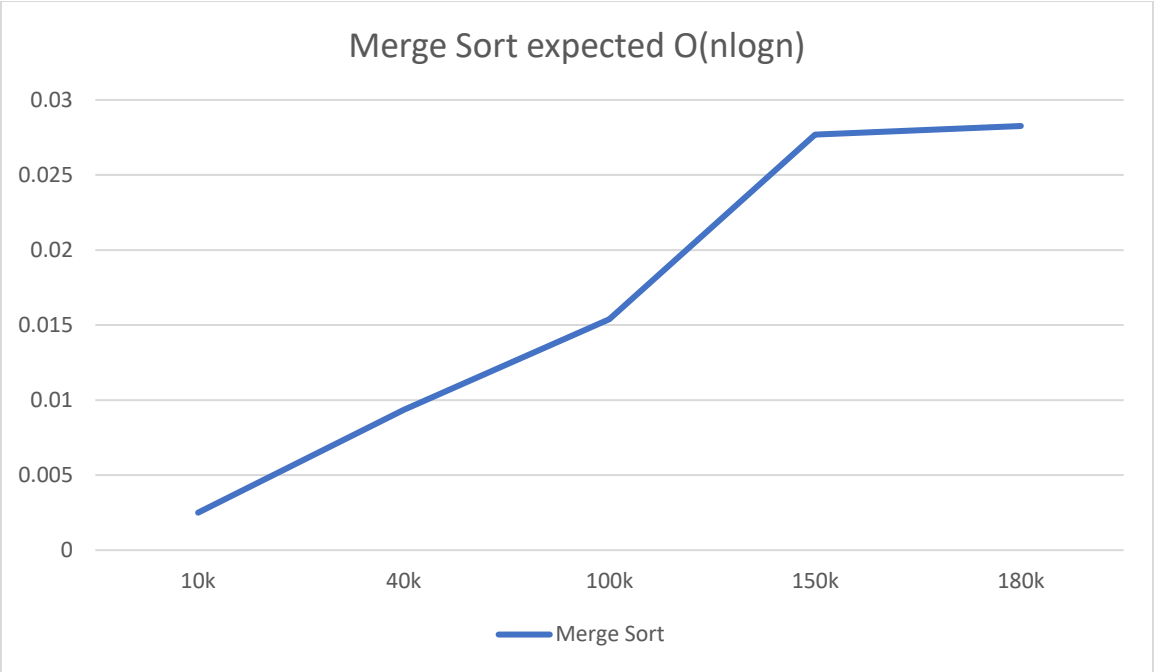
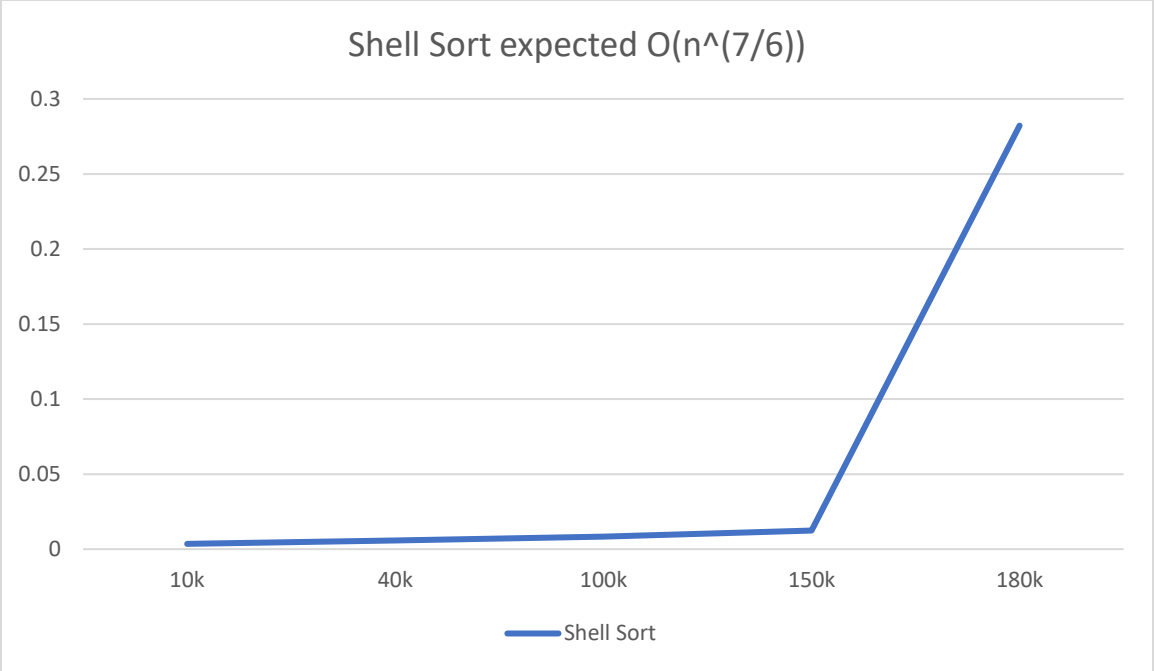


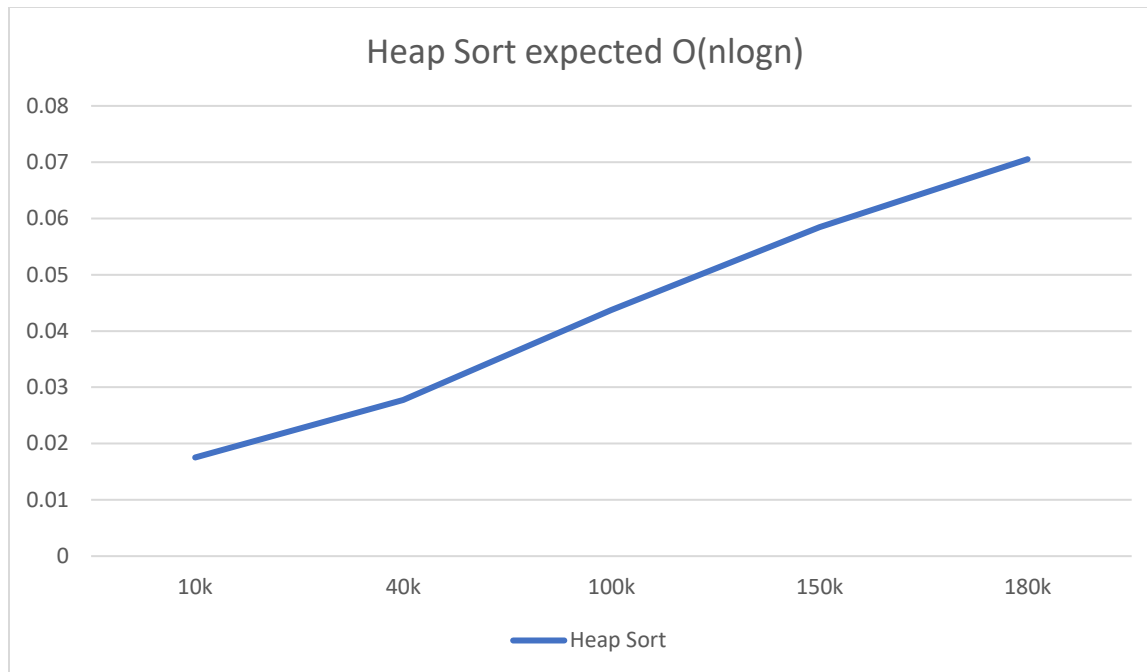


## 5. Individual graphs for sorted arrays









## 6. Individual graphs for sorted linked lists

