Part 4

1.

The worst case Big-Oh complexity of bubble sort is $O(n^2)$. It is because if the array is sorted in a reverse order. Then each time we increase the bubble we would have to iterate the entire bubble due to which it gives $O(n^2)$ time complexity. Meanwhile, the best case Big-Oh complexity of bubble sort is O(n). This is when the array is already sorted correctly. The loop increases the bubble at each iteration and its already sorted. Hence, the inner loop is not run.

2.

The best and worst case of selection Sort is both $O(n^2)$. This is because when the array is sorted or not, in both cases it has to iterate the whole array with outer and inner loop. In both cases it has to iterate to the end of array to find the max/min value. This is why it has n^2 complexity.

3.

Selection Sort does not require any additional storage because it is a stationary sort. It just moves the values in the same array. No extra storage on the heap is used in selection sort.

4.

The big-O complexity would remain the same if we used a linked list instead of an array. This is because we still have to iterate the whole list to find the next smallest or largest number depending on the algorithm. For this reason, the complexity remains the same.

5.

I think Qsort has the complexity of O(nlogn). This is because when we experimented with all sorts on given values. Qsort was the fastest. This hinted that its complexity is faster than O(n), which would be O(nlogn).