

In general, I will approach the problem by looking at the sorting speeds to identify their time complexity and memory usage to identify their space complexity and be able to differentiate between them. I will try to test the algorithms using their best- and worst-case scenarios.

- 1. How will you determine which of the  $n^2$  sorts is bubble sort, which is selection sort, and which is insertion sort? Be specific with input lists you can use to differentiate them.**

Bubble Sort:

- Sorted list - will stop early.
- Reversed list – will be the slowest.

Selection Sort:

- Sorted list - will still take  $O(n^2)$
- List with duplicates - will not preserve original order.

Insertion Sort:

- Sorted list – will be the fastest one.

- 2. How will you determine which quick sort uses which pivot algorithm? Be specific with input lists you can use to differentiate them.**

Last element pivot:

- sorted list - will be the slowest.
- reversed list - will be the slowest.

Random pivot:

- random list - will have different times.

Median Pivot:

- sorted list – will be the fastest one.