

1

Trace six sorts by hand on the following dataset.

Sort the data into ascending order (from smallest to largest).

For each sort, **show the array each time it is altered**. For sorts with a nested loop, show just the result of changes made by the *outer loop*.

Review the lecture notes page 1.9 for examples of how to write your sort traces (<https://insight.ccsf.edu/mod/book/view.php?id=557961&chapterid=154264>).

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

1.

Trace selection sort.

- Indicate in some way (with color or a comment, for example), which elements are being swapped.

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

2.

Trace insertion sort.

- Indicate in some way (with a color, bold, | divider, etc.) what part of the array is the sorted subarray.

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

3.

Trace Shell sort.

- In addition to showing the contents of the array, list what the gap is for that part of the trace.

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

Trace radix sort.

- Show the contents of buckets and the array after each pass.

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

Trace merge sort.

- Show how the arrays are split and how the arrays are merged.

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

Trace Quicksort.

In addition to the trace, submit what the parameters will be for the next two recursive calls to quickSort.

- Use the code given in the else-clause in Section 18.
 - Do not trace a different version of quicksort.
- You only need to show the results after the first partitioning step. In other words, show how the array is changed after the first call to the partition method (displayed in Section 17).
- Note that the partition method includes the call to sortFirstMiddleLast.
- I recommend writing out the variables to help with your trace (e.g., pivotIndex, pivot, indexFromLeft, etc.).

Dataset:

26, 19, 21, 12, 4, 24, 9, 11

Write a method to determine if an array is sorted in ascending order. The method header is:

```
public boolean isSortedAscending(Comparable[] array)
```

For full credit, write a solution that is $O(n)$.

Write a method to determine if a chain of linked nodes is sorted in ascending order. The method header is:

```
public boolean isSortedAscending(Node<Comparable> firstInChain)
```

For full credit, write a solution that is $O(n)$.