1. *What is a “comparison sort?”*

*Answer: It is one of the types of algorithms based on the comparison of two values in one iteration. And the result of this iteration will set the place of one of the elements in the list.*

*What is the best worst-case order of growth for a comparison sort?*

*Answer: n*2

*What is the best worst-case order of growth for any sort algorithm?*

*Answer: n \* n!*

1. *What is the order of growth of bubble sort, and why does Barack Obama think it is “the wrong way to go?”*

*Answer: The worst case for Bubble sort is n*2 *. It is the wrong way to go for a huge size input because there are more efficient algorithms than this one.*

1. *What is the order of growth of radix sort?*

*Answer: O(nk) (k - number of digits the biggest word or digit).*

*What preconditions do we need to use it?*

*Answer: Implementation of radix sort works with letters or digits so the sequence has to be one of those types to get a reasonable result.*

1. *What is a stable sort and why might it matter in practice?*

*Stable sort saves the order of two identical elements of the list. It may be important when the data is complex and sort algorithms are applied to different parts of the data.*

1. *What is the worst sorting algorithm (that has a name)?*

*Bogosort*

1. *What sort algorithm does the C library use?*

*Answer: Quick sort*

*What sort algorithm does Python use?*

*Answer: Timsort (based on insertion sort and merge sort)*

*Are these algorithms stable? You might have to Google around to find these answers.*

*Answer: Quick sort is not stable. Timsort is stable.*

1. *Many of the non-comparison sorts are linear, so why does Python use an O*(*n* log*n*) *comparison sort?*

*Answer: It is a stable algorithm and works on real-time data. In this, the list that needs to be sorted is first analyzed and based on that the best approach is selected.*