Sorting Worksheet

1. A new sorting algorithm can sort elements by allocating an array that is 1/3 the size of the array that needs to be sorted, and will run faster when data is not sorted. Elements with the same value might be swapped as needed. Which of the following properties apply to this algorithm?

    a. In-place
    b. Internal
    c. Adaptive
    d. Stable
    e. None of the above

2. A graduate student has come up with a new comparison-based sorting algorithm. The student claims the big O of the algorithm is O(k x n) where k is a constant between 1 and 10. What could you say about this algorithm? Is this algorithm possible? Briefly discuss.

3. The bubble sort code presented in lecture can be optimized. Mention the optimization and rewrite the code.

4. You have a small set of data to be sorted. Which quadratic algorithm would you use?

5. Both quicksort and mergesort have O(n(log(n)) algorithmic complexity. Why would you use one over the other? Explain.

6. If you are using Radix sort, why many buckets will be required to:

    a. Sort binary numbers
    b. Sort hexadecimal numbers

7. In Radix sort we process digits right to left; would it make a difference if we process digits left to right? Explain.

8. Which sorting algorithms always have the same worst and average complexity?

9. Consider how would you implement a recursive version of:

    a. Selection sort
    b. Insertion sort
    c. Bubble sort

10. Insertion sort is $O(n^2)$ in the worst case, but it is O(n) in the best case. When does the best case take place?

11. In Quick sort, picking the median as the pivot is ideal. Why?

12. Python is a cool language ☺

13. Which of the following partitions are valid partitions generated by the partition method associated with Quick sort? The bolded/underlined element represents the pivot.

    a. 5 15 3 **<u>11</u>** 20 9
    b. 5 **<u>15</u>** 3 11 20 9
    c. 5 9 3 **<u>11</u>** 20 15
    d. **<u>3</u>** 9 5 11 20 15