1. **Divide-and-conquer.** The divide-and-conquer paradigm can be described in general terms as consisting of the following three steps: (1) divide, (2) recur, and (3) conquer.

Write down the recurrence relation showing the running time of a generic recursive, divide-and-conquer algorithm:

\[ T(n) = D(n) + kT\left(\frac{n}{2}\right) + C(n) \]  

2. Suppose a recursive divide-and-conquer algorithm partitions the original problem into 2 subproblems of size \(n/4\) and 3 subproblems of size \(n/6\), and that it spends \(\Theta(n)\) time to partition the original problem into subproblems and \(\Theta(n \log n)\) time to combine the subproblem solutions into the solution to the original problem, where \(n\) is the input size.

Write down the recurrence relation showing the running time of this algorithm:

\[ T(n) = 2T\left(\frac{n}{4}\right) + 3T\left(\frac{n}{6}\right) + \Theta(n \log n) \]  

3. **Quick Sort.** Quick sort is an algorithm that applies the divide-and-conquer paradigm. Quick sort partitions the original problem of size \(n\) into 2 subproblems and spends \(O(n)\) time total to partition the problem into subproblems and to combine the subproblem solutions into a solution to the original problem. The overall running time of quick sort is \( O(n \log n) \) time (average case).

Write down the recurrence relation showing the running time of quick sort (for the average/expected case):

\[ T(n) = 2T\left(\frac{n}{2}\right) + O(n) \]  

4. **Bucket Sort.** Bucket sort differs from insertion sort, selection sort, heap sort, merge sort, and quick sort because it does not use comparison comparisons.

Assuming there are \(n\) elements to be sorted and that the bucket sort uses \(N\) buckets, the running time of bucket sort is \( O(n + N) \).

5. True or False: Bucket sort is a stable, in-place sort.

6. **Radix Sort.** Radix sort internally applies Bucket sort to lexicographically order a set of tuples. Assuming each of the \(n\) tuples have \(d\) dimensions and there are \(N\) buckets, the running time of radix sort is \( O(d(n + N)) \).

7. \(\check{\text{True}}\) or False (Circle one): Radix sort is a stable sort and is not in-place.