Problem 1. (40 points)
1. Briefly describe how mergesort algorithm works.
   Mergesort recursively divides an array down to the base case of array size = 1. Then each subarray is sorted, then combined recursively with the other sorted subarrays until the entire array is sorted. Mergesort requires an auxiliary array as extra space to accomplish sorting.

2. Given an array of integers: 3, 5, 9, 2, 6, 8, 1, 7, show the key intermediate steps of how the array gets sorted.
   Divide:
   
   \[
   \begin{array}{c}
   3 | 5 | 9 | 2 | 6 | 8 | 1 | 7 \\
   \end{array}
   \]

   Conquer:
   
   \[
   \begin{array}{c}
   3, 5 | 2, 9 | 6, 8 | 1, 7 \\
   2,3,4,9 | 1,6,7,8 \\
   1,2,3,5,6,7,8,9 \\
   \end{array}
   \]

3. The runtime cost for mergesort is given as the following recurrence relation.
   \[
   T(N) = 2T(N/2) + N \\
   T(1) = 0
   \]
   Solve the recurrence relation, and derive the Big-O notation.

\[
N = 2^k \quad k = \log N \\
T(2^k) = 2T(2^{(k-1)}) + 2^k \\
= \ldots \\
= 2^k T(2^0) + k2^k \\
= 0 + k2^k \quad = O(N \log N)
\]

Problem 2. (30 points)
1. Briefly describe how quicksort algorithm works.
   Quicksort shuffles the array and partitions the array using a pivot value. The array is recursively sorted so that all elements less than the pivot are on the left and those greater are on the right. This is an in-place sort so no extra space is required.

2. Given an array of integers: 3, 5, 9, 2, 6, 8, 1, 7, show the key intermediate steps of how the array gets sorted.

   pivot = 3: 2, 1 3 5,9,6,8,7
   pivot =2: 1,2,3,5,9,6,8,7
   ...
   pivot =9: 1,2,3,5,6,7,8,9

3. What’s the average runtime cost of quicksort in big O notation? (You do not have to show the recurrence relation and intermediate steps.)
   \(O(2N \log N)\)
Problem 3. (30 points)
1. Briefly describe how binary search algorithm works given a sorted array.
   The algorithm picks a middle element from a sorted array and compares that to the search key.
   If the key is smaller, only the left subarray contains the key. If larger, only the right subarray.
   The algorithm searches until either an answer is found or no elements remain to be searched.

2. Given a sorted array of integers: 1, 2, 3, 4, 5, 6, 7, 8, show the comparisons that have to be done if the search key is 3.

   4 > 3
   2 < 3
   3 = 3 Done!

3. What’s the cost function in recurrence relation for binary search? (You do not have to solve it.)
   \[ T(N) = T(N/2) + 1 \]