Problem 1. Use heap sort algorithm, shown in class, to sort the following array, $A = [4, 3, 2, 8, 1, 9]$, and answer the following questions:

1. Show heap and array at each step.
2. Count the exact number of comparisons at each step.
3. Count the exact number of exchanges at each step.
4. What is the total number of comparisons to sort, A, using heap sort algorithm?
5. What is the total number of exchanges to sort, A, using heap sort algorithm?

Show your work.

Problem 2. For an arbitrary array of size, n, solve for the exact number of exchanges that take place in sorting this array using heap sort algorithm. Show your work for exact analysis in the worst case.

Problem 3. You are given an unordered input array, $X = [x_1, x_2, \ldots, x_n]$ of length, n and another sequence, $A = [a_1, a_2, \ldots, a_n]$ of distinct integers from 1 to n, such that, $a_1, a_2, \ldots, a_n$ is a permutation of $1, 2, \ldots, n$. Design an efficient algorithm and write pseudo-code to order X according to the order imposed by the permutation. For example, if $X = [27, 15, 11, 19]$, and $A = [3, 2, 4, 1]$ then the output should be $X = [19, 15, 27, 11]$. The algorithm should be in-place, so you cannot use an additional array. What is the runtime of your algorithm.

Problem 4. Suppose there is an array of length n. It contains values in the range of $[1 \ldots n + 1]$. However, exactly one value out of $\{1, \ldots, n + 1\}$ is missing from this array. Find this missing number as efficiently as possible when the array is sorted such that the value stored at *index* $1 <$ value at *index* $2 < \ldots <$ value at *index* n. Write pseudo code to find the missing number. Analyze the algorithm exactly.