Assignment 3, Due: March 8.

1. Show that for any comparison based sorting algorithm, the average number of comparisons is $\Omega(n \log n)$.

2. Write an efficient algorithm to find k smallest elements in an array. Input: Array and integer k.

3. Recall, finding i^{th} minimum algorithm, what would happen to the analysis if we choose group size to be 3 or 7. Between group size 5 and 7, which one is best.

4. Write an algorithm to find maximum and second maximum in $n + \log n - 2$ comparisons.

5. What is the best and average time complexity of Heap sort.

6. Suppose a given array is represented as a linked list, among the sorting algorithms discussed (bubble, insert, merge, quick, heap), which all can be easily modified to work for linked list. If it can not be modified, justify why. Also, analyse the time complexity in worst case if implementation is possible.

7. You are given a set of strings and asked to arrange them in lexicographic order (dictionary order), which sorting algorithm is good for this application. What would be the lower bound for any sorting algorithm in the worst case.

8. `CLRS Ch 8-4: Water Jug`. Suppose that you are given n red and n blue water jugs, all of different shapes and sizes. All red jugs hold different amounts of water, as do the blue ones. Moreover, for every red jug, there is a blue jug that holds the same amount of water, and vice versa. It is your task to find a grouping of the jugs into pairs of red and blue jugs that hold the same amount of water. Allowed operation: pick a pair of jugs in which one is red and one is blue, fill the red jug with water and then pour the water into the blue jug. This will tell you whether the jugs are of same volume or which one is bigger and assume this comparison is one time unit. Goal is to make a minimum number of comparisons to determine the grouping. Present an efficient algorithm with analysis. What is the lower bound for this problem.