You're expected to work on the problems before coming to the lab. Discussion session is not meant to be a lecture. TAs will guide the discussion and correct your solutions if needed. We may not release solutions for all problems. If you're better prepared for discussion, you will learn more. TAs will record names of the students who actively engage in discussion and report them to the instructor. The instructor will factor in participation in final grade.

1. (Intermediate) Draw the decision tree for Merge sort operating on three element; so Merge-sort(A, 1, 3) where the input is $A[1...3] = \langle a_1, a_2, a_3 \rangle$. Use the decision tree for Insertion sort in Fig. 8.1 as a model. Note that each internal node must correspond to a comparison of two given elements from $A[1...3]$.

2. (Basic) What is a stable sort?

3. (Basic) Explain why the decision tree for any comparison based sorting algorithm must have at least $n!$ leaves.

4. (Basic) Show $\log_2 n! = \Omega(n \log n)$.

5. (Basic) Illustrate the operation of Counting-Sort on $A = \langle 6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2 \rangle$.

6. (Advanced) Describe an algorithm that, given n integers in the range from 0 to k, preprocesses the input and then answers any query on how many of the n integers fall into range $[a...b]$ in $O(1)$ time. Your algorithm should use $O(n + k)$ preprocessing time.

7. (Intermediate) Which of the following sorting algorithms are stable: insertion sort, merge sort, and quicksort (according to their implementations in the textbook)?

8. (basic) Using Figure 8.4 as a model, illustrate the operation of Bucket-Sort on the array $A = \langle .79, .13, .16, .64, .39, .20, .89., 53., .71, .42 \rangle$.

9. (intermediate) Explain why the worst-case running time for bucket sort is $\Theta(n^2)$? What simple change to the algorithm preserves its linear average-case running time and makes its worst-case running time $O(n \log n)$?

10. (advanced) Describe a (worst-case) linear time algorithm that decides if a given sequence of integers $\langle a_1, a_2, ..., a_n \rangle$ is a permutation of $\langle 1, 2, 3, ..., n \rangle$. Your algorithm only needs to say Yes or No. (Hint: counting sort.)