You are expected to work on the problems before coming to the lab. Discussion sessions are not meant to be a lecture. TA will guide the discussion and correct your solutions if needed. We will not release "official" solutions. If you are 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 factors-in participation in the final grade as bonus points.

1. Order the following functions in asymptotically non-decreasing order using $\leq$ or $=$[1].
   $n^n, 2^n - n^2, \lg n, \lg\lg n, 7n^3 + 10n^2, n^3 \lg n, 3^n, n \lg n, n^{\lg n} 50000$.

2. (Basic) Rank the following functions by order of growth; that is, find an ordering $g_1, g_2, \cdots, g_k$ (here k is the number of functions given) such that $g_1 = O(g_2)$, $g_2 = O(g_3), \cdots, g_{k-1} = O(g_k)$. (For example, if you are given functions, $n^2, n, 2n$, your solution should be either $n, 2n, n^2$ or $2n, n, n^2$.)

$n^2 + 2^n$	$n \log n$	$n^2 \log^2 n$	$n \log^2 n$	n^2	$\log^{100} n$
$\log\log n$	n^3	1	$\log n$	$\log n / \log\log n$	$n^2 / \log n$

3. (Basic) You're given an array $A[1 \cdots 8] = \langle 3, 1, 6, 8, 4, 2, 5, 7 \rangle$. Is this a max-heap? If not, make it a max-heap by iteratively applying the Max-Heapify function. In other words, illustrate the operation of Build-Max-Heap on the array.

4. (Basic) Illustrate the operation of Max-Heap-Insert(A, 9) on the heap you obtained in problem 1.

5. (Basic) Illustrate the operation of Heap-Sort on the max-heap you've obtained in problem 2. In other words, you only need to execute Lines 2– of Heap-Sort on the current heap.

6. (Basic) You're given an array $A[1 \cdots 8] = \langle 3, 1, 6, 8, 4, 2, 5, 7 \rangle$. Is this a min-heap? If not, make it a min-heap by iteratively applying the Min-Heapify function.

7. (Basic) Is an array that is sorted order (increasing order) a min-heap?

8. (Basic) Is Heap-Sort an in-place sorting algorithm? We say that a sorting algorithm is in-place if only $O(1)$ elements are stored outside the input array at any time. Do you see why Insertion-Sort is in-place but Merge-Sort is not?

9. (Intermediate) You want to add a new operation to the max-priority-queue, namely Heap-Second-Maximum(A). The function returns the second largest value (key) stored in the heap A. Give a pseudo-code of this operation. What's the running time of your algorithm? The faster, the better.

[1] For example, $n \leq 3n^2 + 5 = n^2 - 3 \leq n^3$.

10. (Advanced) The operation Heap-Delete(A, i) deletes the item in node i for heap A. Give an implementation of Heap-Delete that runs in $O(\lg n)$ time for an n-element max-heap.

11. (Advanced) Give an $O(n \lg k)$-time algorithm to merge k sorted lists into on sorted list, where n is the total number of elements in all the input lists. Hint: Use a min-heap.

12. (Basic) Using Figure 7.1 as a model, illustrate the operation of Partition on the array $A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 21, 2 \rangle$.

13. (Basic) Is Quick-Sort an in-place sorting algorithm?