CSE100 Algorithm Design and Analysis Discussion Ch 04 Spring 2019
Last update: 1/30/2019

You're expected to work on the problems before coming to the lab. Discussion session is 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'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. (basic) Consider the pseudo-code of Merge-Sort with line 1 removed. What goes wrong?
2. (advanced) Let's slightly tweak Merge Sort. Instead of partitioning the array into two subarrays, we will do into three subarrays of an (almost) equal size. Then, each recursive call will sort each subarray, and we will merge the three sorted subarrays. What is the running time of this new merge? Write a recurrence for the running time of this new version of Merge Sort, and solve it.
3. (basic) Solve $T(n) = 4T(n/2) + \Theta(n)$ using the recursion tree method. Cleary state the tree depth, each subproblem size at depth d, the number of subproblems/nodes at depth d, workload per subproblem/node at depth d, (total) workload at depth d.
4. (basic) Solve $T(n) = 4T(n/2) + \Theta(n^2)$ using the recursion tree method.
5. (basic) Solve $T(n) = 4T(n/2) + \Theta(n^3)$ using the recursion tree method.
6. (basic) Solve the above recursions using the Master Theorem. You must specify which case and how you set the variables such as a, b, ϵ.
7. (basic) Solve $T(n) = 2T(n/2) + 1$ using the substitution method. You just need to give an asymptotic upperbound (preferably tight, though).