CS2040S: Data Structures and Algorithms

Tutorial Problems for Week 4

For: Sep 2–6, 2019

Problem 1. Top-k
In an earlier lecture, we discussed the Top-k problem, i.e., finding the largest/smallest k items in a list. During the discussion, the Quickselect algorithm was (perhaps) mentioned and we'll discuss this cool algorithm during this tutorial. A key step of the Quickselect (and Quicksort) algorithm is the partition function which we have covered in lecture. First, a couple of warm-up questions.

Problem 1.a. What are the two partition functions that we learnt in lecture? Briefly describe how both methods work and provide pseudocode.

Problem 1.b. What is the computational complexity of the two partitioning methods?

Problem 1.c. We can use quicksort with the partition function to find the k-th smallest element in a list in $O(n \log n)$ time. How can we achieve this?

Problem 1.d. Now, think about how you can use modify Quicksort to find the k-th smallest element in a list. *Provide pseudocode for your algorithm and analyze its worst-case time complexity.*

Problem 1.e. Next, modify your algorithm to find the top-k elements. Does it matter if we need these elements to be sorted or not? What is the worst case time complexity of your method? Can you make a guess about its average time complexity?