Problem 1. Suppose that the splits at every level of quicksort are in the proportion $1 - \alpha$ to α, where $0 < \alpha \leq 1/2$ is a constant. Write the recurrence equation. Show that the minimum depth of a leaf in the recursion tree is $-lg\, n/lg\, \alpha$ and the maximum depth is approximately $-lg\, n/lg\, (1 - \alpha)$.

Problem 2. In class we did different cases of Quicksort algorithm for various splits of the input data based on a choice of the pivot. For this problem we are going to assume that a pivot is selected such that data is partitioned in the ratio of 2 to 1 every time. The partition routine would remain the same as used in class and so would the number of comparisons in it. Answer the following questions:

(a) Write the recurrence equation, and the base case.

(b) What is the height of the recursion tree?

(c) Solve the recurrence equation using an appropriate method. Justify your method.

(d) Verify the base case.

Problem 3. Suppose I want to find the k-th largest number in an array of size n. I could sort the array and look at the k-th value from the end. This could be an $O(nlg\, n)$ runtime algorithm. We would like to improve it. Write an algorithm in English or in pseudo-code to find the k-th largest value in $O(klg\, n)$ runtime for large k. As an example, the 3^{rd} largest value in, $A = [4, 2, 3, 1, 6, 8]$, is 4.

Note: For smaller values of k, the runtime is obviously linear. For this problem we are seeking the runtime when k is closer to n than to 1, such that, $1 << k \leq n$.