Collaboration is encouraged, but students MUST write up solutions independently and list those with whom they worked. No materials from prior years’ classes or the Internet can be consulted.

1. (20 pts) Solve the following recurrence relations. Find both upper and lower bounds. Assume that $T(0) = 1$ and $T(1) = 1$. Justify your answers and make your bounds as tight as possible.
   
   (a) $T(n) = 10T(\log n) + n$
   
   (b) $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

2. (20 pts) Suppose that at each run of Quicksort, the splits are in the proportion $1 - \alpha$ to $\alpha$ for some constant $0 < \alpha < 1/2$. Show that the minimum depth of a leaf of the recursion tree is approximately $-\log n / \log \alpha$, and that the maximum depth is approximately $-\log n / \log (1 - \alpha)$.

3. (20 pts) Give an algorithm that, given $n$ integers in the range 0 to $k$, preprocesses its input and then answers any query about how many of the $n$ integers fall into any range $[a...b]$ in $\Theta(1)$ time. Your algorithm should use $\Theta(n + k)$ preprocessing time.

4. (20 pts) Suppose that we have an infinitely long array of bits. The array is filled with 0’s from index 1 to index $k$, and is filled with 1’s for all indices greater than $k$. Give an algorithm that finds $k$ in $O(\log k)$ time.

5. (20 pts) Suppose that we have a function named $\text{successor}$ that takes a node as an input and finds the next largest item in its binary search tree. Prove that no matter what node we start at in a height-$h$ binary search tree, $k$ successive calls to $\text{successor}$ take $O(k + h)$ time.