Homework 2
ITCS-6114/8114: Algorithms and Data Structures

Due: Thursday, September 23, 2021

Homeworks are due at the **beginning of class** on Thursday, September 23. **Late homeworks will receive no credit.** Homeworks are to be done individually and will be graded on the basis of correctness, clarity, and legibility. Show the steps in your work. Each question is worth **10 points**, for a total of **50 points**.

Be sure to write your name, student ID, and email address on your homework submission.

1. (a) Suppose the splits at every level of quicksort are in the proportion $1 - \alpha$ to α, where $0 < \alpha < 1/2$ is a constant. Show that the minimum depth of a leaf in the recursion tree is approximately $-(\log n)/(\log \alpha)$ and the maximum depth is approximately $-(\log n)/(\log(1 - \alpha))$. (Do not worry about integer roundoff.)

   (b) Use the recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1 - \alpha)n) + cn$, where $c > 0$ is a constant.

2. You are given an array of strings, where different strings may have different numbers of characters, but the total number of characters over all the strings in the array is n. Show how to sort the strings in $O(n)$ time.
   (Note that the desired order here is the standard alphabetical order; for example, `a < ab < b`.)

3. Nuts and bolts: You are given a collection of n bolts of different widths and n corresponding nuts. You are allowed to try a nut and bolt together, from which you can determine whether the nut is larger than the bolt, smaller than the bolt, or matches the bolt exactly. However, there is no way to compare two nuts together or two bolts together. The problem is to match each bolt to its nut. Design an algorithm for this problem with an average-case efficiency of $\Theta(n \log n)$.

4. You are given two sorted lists of size m and n. Give an $O(\log m + \log n)$ time algorithm for computing the kth smallest element in the union of the two lists.

5. (a) Show the contents of the hash table after inserting the keys S, T, R, U, C, T, U, R, E (in that order) into a hash table of length $m = 13$ for hashing with chaining. Let the hash function be $h(k) = (\text{position of } k \text{ in the alphabet}) \mod m$. For example, the letter A has position 1, B has position 2, and S has position 19.

   (b) Show the contents of the hash table after inserting the keys S, T, R, U, C, T, U, R, E (in that order) into a hash table of length $m = 13$ using open addressing with linear probing. Let the hash function for linear probing be $h(k,i) = (h^{'}(k) + i) \mod m$, where the auxiliary hash function $h'(k) = (\text{position of } k \text{ in the alphabet}) \mod m$. For example, the numerical value of the key A is 1, key B is 2, and key S is 19.

(c) Show the result of inserting the keys S, T, R, U, C, T, U, R, E (in that order) into a hash table of length $m = 13$ using open addressing with double hashing. Let $h_1(k) = k \mod m$ and $h_2(k) = 1 + (k \mod m^{'})$ where $m^{'} = 11$. Again assume the numerical value of an alphabetical key k is its position in the alphabet.