CS515: Algorithms and Data Structures, Fall 2016

Homework 3*

Due: Tue, Nov/15/16

Homework Policy:

1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. Of course, you can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. More items might be added to this list. ☺

Problem 1. [**40 pts**] Suppose you are given n bolts and n nuts. The bolts have distinct sizes. Also, the nuts have distinct sizes. Each bolt match exactly one nut.

Consider the following randomized algorithm for choosing the **largest** bolt. Draw a bolt uniformly at random from the set of n bolts, and draw a nut uniformly at random from the set of n nuts. If the bolt is smaller than the nut, discard the bolt, draw a new bolt uniformly at random from the unchosen bolts, and repeat. Otherwise, discard the nut, draw a new nut uniformly at random from the unchosen nuts, and repeat. Stop either when every nut has been discarded, or every bolt except the one in your hand has been discarded.

(a) What is the probability that the algorithm discards NO bolts?

(b) What is the exact expected number of discarded bolts when the algorithm terminates?

(c) What is the probability that the algorithm discards exactly one nut?

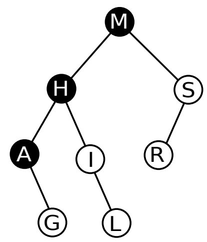
(d) What is the exact expected number of discarded nuts when the algorithm terminates?

(e) What is the exact expected number of nut-bolt tests performed by this algorithm?

*Problems are from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on randomization is recommended.

Problem 2. [**40 pts**] Consider a randomized treap T with n vertices.

(a) The left spine of a binary tree is a path starting at the root and following left-child pointers down to the first node with no left child pointer. What is the expected number of nodes in the left spine of T? For example, the number of nodes on the left spine of the following tree is 3, all black nodes. [Hint: What is the probability that a node is on the left spine.]



(b) What is the expected number of leaves in T? [Hint: What is the probability that a node is a leaf?]

Problem 3. [**20 pts**] Suppose that the worst case expected running time of an algorithm on an input of size n is given by $T(n)$. Find an upper bound on the probability that the algorithm takes more than $10 \cdot T(n)$ on any given input. [Hint: study Markov inequality]