EE360C: Algorithms
The University of Texas at Austin — Exam #1 Review
Dr. David Soloveichik — October 3, 2018

Name: **EID:**

Exam #1 Review

Instructions. WRITE LEGIBLY.
No calculators, laptops, or other devices are allowed. This exam is **closed book**. Write your answers on the test pages. If you need scratch paper, use the back of the test pages, but indicate where your answers are. Write down your process for solving questions and intermediate answers that **may** earn you partial credit.

If you are unsure of the meaning of a specific test question, write down your assumptions and proceed to answer the question on that basis.

When asked to describe an algorithm, you may describe it in English or in pseudocode. If you choose the latter, make sure the pseudocode is understandable.

If you write information in response to a question, and that information is incorrect, you will not earn full credit. In the same vein, if a question asks for a finite number of things, and you provide "extra" things, we will ignore anything extra, and grade only the first answers.
You have **90 minutes** to complete the exam.
Some useful information:

Problem 1: Stable Job Offers

Consider a workforce economy made up of n available jobs and n people looking for work, where n is an even positive integer. Each worker has a list of preferences for jobs, and each job ranks all of the available workers. There are no ties in these lists. Half of the jobs are full-time jobs and half of the jobs are part-time jobs. We assume that every worker prefers any full-time job over any part-time job. Further, half of the workers are inherently "hard workers" and the other half are inherently "lazy". We assume that every employer prefers any hard worker over any lazy worker. We assume a definition of stability identical to what we proved for stable marriage: a matching of workers to jobs is stable if there are no instabilities (i.e., there does not exist two pairs (j, w), (j', w') such that the employer offering job j prefers worker w' to w and w' also prefers job j to j'). Prove that, in every stable matching of workers to jobs, every hard worker gets a full time job.

Problem 2: Asymptotic Notation

Prove or disprove each of the following. You may use either the definitions of the asymptotic notations or the limit method.

(a) $f(n) = O(g(n))$ implies $g(n) = O(f(n))$

(b) $f(n) + g(n) = \Omega(\min(f(n), g(n)))$

Problem 3: Heap Algorithm Design

Let W be an unsorted array of distinct elements and S be W sorted. We say W is k-wrong if for all i, $S[i] = W[j]$ implies $|i - j| < k$. I.e., each element in W is at most k positions away from its correct, sorted position. Give an algorithm using a heap to sort W in $\mathcal{O}(n \log k)$ time.

Problem 4: Depth First Search

During the execution of depth first search, we refer to an edge that connects a vertex to an ancestor in the DFS-tree as a *back edge.* Either prove the following statement or provide a counter-example: if G is an undirected, connected graph, then each of its edges is either in the depth-first search tree or is a back edge.

Problem 5: Counting Shortest Paths

In addition to the problem of computing a single shortest v-w path in a graph G, we are interested in the problem of determining the *number* of shortest v-w paths.

This turns out to be a problem that can be solved efficiently. Suppose we are given an undirected graph $G = (V, E)$, and we identify two nodes v and w in G. Give an algorithm that computes the number of shortest v-w paths in G. (The algorithm should not list all the paths; just the number suffices.) The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.