**CSE 347 Analysis of Algorithms** May 2, 2017

# Sample Final Exam

*Instructors: Brendan Juba & Robert Utterback* *2 hours*

This is a sample final exam. It is intended to be representative of the content, length, and difficulty of the questions you will be asked on your final exam.

**Instructions.** Answer questions as clearly and concisely as you can. Try to fit your solutions in the space provided, continuing on the provided extra sheets of paper if necessary. We will award partial credit based on the work that you provide; if you get stuck, *be sure to explain what you are trying to do.*

This is a *closed book* exam. The use of laptops, phones, smartwatches, etc. is *not allowed.* You are allowed to reference a crib sheet that you prepared yourself, contained entirely on a single 8.5×11" (US letter size) sheet of paper. Any other written material is not permitted.

You may use (without proof) any theorem that has been stated in lecture. Except on the Reading Solutions problem, you may also reference the result of the homework problems you were assigned without proof. You *may not* use the result of other problems from the book without providing a proof.

**Name:**

**Student ID:**

| Problem Number | Points Possible | Points Received |
|---|---|---|
| 1 | 10 | |
| 2 | 15 | |
| 3 | 20 | |
| 4 | 20 | |
| 5 | 25 | |
| 6 | 25 | |
| TOTAL | 115 | |

1. (a) *(5 points)* Suppose we have a randomized algorithm for an *optimization problem* that only produces an optimal solution with some small probability $\epsilon > 0$. (We assume that we can compute the objective value of the returned solutions in polynomial time using a deterministic algorithm.) Describe how to obtain a polynomial-time algorithm that returns a correct solution with probability $1 - \delta$ for a given $\delta > 0$ (and show that your solution is correct).

   (b) *(5 points)* Suppose we have a randomized algorithm for a (yes/no) *decision problem* that only returns the correct decision with probability $1/2 + \epsilon$ for some small $\epsilon > 0$. In this case, we *do not* have an algorithm to check whether or not the returned answers are correct. Describe how to obtain a polynomial-time algorithm that returns the correct answer with probability $1 - \delta$ for a given $\delta > 0$ (and show that your solution is correct).

2. *(15 points)* **Reading solutions.** The following is a problem that appeared on your homework. *For this problem only,* you may not simply reference the fact that it appeared on your homework. You must present a solution to this problem.

   Let $G = (V, E)$ be an undirected graph with $n$ nodes and $m$ edges. For a subset $X \subseteq V$, we use $G[X]$ to denote the subgraph *induced* on $X$ – that is, the graph whose node set is $X$ and whose edge set consists of all edges of $G$ for which both ends lie in $X$.

   We are given a natural number $k \leq n$ and are interested in finding a set of $k$ nodes that induces a "dense" subgraph of $G$; we'll phrase this concretely as follows. Give a polynomial-time algorithm that produces, for a given natural number $k \leq n$, a set $X \subseteq V$ of $k$ nodes with the property that the induced subgraph $G[X]$ has at least $\frac{mk(k-1)}{n(n-1)}$ edges.

   You may give either (a) a deterministic algorithm, or (b) a randomized algorithm that has an expected running time that is polynomial, and that only outputs correct answers.

3. *(20 points)* Your dorm room is in the center of a very long hallway, with $n/2$ evenly-spaced rooms on either side. One of these rooms has free pizza, but you don't know which one. All the doors are open, so you only need to walk past a room to decide if it contains the free pizza. We want to minimize the number of rooms we must walk past before finding the free pizza. Give a 9-competitive algorithm for this problem.

4. *(20 points)* Consider a telecommunications network consisting of a cycle on $n$ nodes, numbered 0 through $n-1$ clockwise around the cycle. Some set $C$ of calls is given; each call is a pair $(i, j)$ originating at node $i$ and destined for node $j$. The call can be routed either clockwise or counterclockwise around the ring. Our goal is to route the calls so as to minimize the total *load* on the network. The load $L_i$ on link $(i, i+1 \pmod n)$ is the number of calls routed through link $(i, i+1 \pmod n)$, and the total load is

$$max_{1 \leq i \leq n} L_i$$

Give a 2-approximation algorithm for this problem.

5. *(25 points)* Recall the definition of a spanning tree: given an undirected graph $G = (V, E)$, a spanning tree is a connected subgraph $T = (V, E')$, where $T$ is a tree and $E' \subseteq E$.

   Consider the following problem. Given as input a pair $(G, k)$ where $G$ is an undirected, connected graph and $k$ is an integer, decide whether $G$ contains a spanning tree such that all vertices in the tree have degree *at most* $k$?

   Either prove this problem is NP-Complete or give a polynomial time algorithm to solve it.

6. *(25 points)* Consider the following problem. Given an input pair $(G, L)$ where $G$ is an undirected, connected graph and $L \subseteq V$ is a set of vertices in $G$, decide whether $G$ contains a spanning tree $T$ in which every vertex in $L$ is a leaf in $T$ ($T$ may contain other leaves in addition).

   Either prove this problem is NP-Complete or give a polynomial time algorithm to solve it.