# Qualifying Examination
## Theoretical Computer Science
Saturday, February 28, 2009

## Part I: Algorithms

| | |
|---|---|
| **ID Number** | |
| **Pseudonym** | |

| Problem | Maximum Points | Points Earned | Grader |
|---|---|---|---|
| 1 | 25 | | |
| 2 | 25 | | |
| 3 | 25 | | |
| 4 | 25 | | |
| Total | 100 | | |

**Instructions:**

1. This is a closed book exam.
2. The exam is for 3 hours and has four problems of 25 points each. Read all the problems carefully to see the order in which you want to tackle them.
3. Write clearly and concisely. You may appeal to some standard algorithms/facts from text books unless the problem explicitly asks for a proof of that fact or the details of that algorithm.
4. If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half baked ideas just to get partial credit.

May the force be with you.

**Problem 1:** A bus driver and his bus begin at location 0 on a straight road, with $n$ of passengers. Passenger $i$ would like to get off at location $L_i$ given by a rational number (positive indicates $L_i$ is to the right of the bus, negative is to the left of the bus). The driver drops off the passengers by driving back and forth along the road according to whatever strategy he adopts. For example, he might choose to drive forward in the positive direction, dropping off passengers as he goes, until he reaches the passenger with largest $L_i$, then turn around and go in the negative direction until the passenger with least $L_i$ gets off. Alternatively, he might alternate forward and back, zig-zagging many times and changing directions each time he drops off a passenger.

The driver is charged a gas penalty of one unit for each mile each passenger travels on the bus. For example, suppose that the bus is at location 0, and that there are passengers who wish to get off at locations 3,4, and $-2$. If the driver drops off the passengers in this order: 3,4,$-2$, then the total gas used is $3 + 4 + 10 = 17$, but if the driver drops off the passengers in this order: 3,$-2$,4, then the total gas used is $3 + 8 + 14 = 25$.

Bus-driver problem: Given a bus at location 0, and $n$ passengers on the bus who wish to get off at locations $L_1, \ldots, L_n$, what is the least amount of gas needed in order to drop off all of the passengers? Either prove that the bus-driver problem is NP-hard, or give a polynomial-time algorithm for solving it.

**Problem 2:**

1. Give a proof of Euler's formula for planar graphs: Any planar embedding of any connected planar graph with $V$ vertices, $E$ edges, and $F$ faces (including the outer face) satisfies the identity $V - E + F = 2$.

2. Give a proof of Euler's formula for toroidal graphs: Any embedding of any connected graph onto the torus with $V$ vertices, $E$ edges, and $F$ faces, *each of which is a topological disk*, satisfies the identity $V - E + F = 0$.

Your proofs should be as elementary and self-contained as possible. Clearly point out which nontrivial topological results (like the Jordan curve theorem) your proof requires.

**Problem 3:** Let $G = (V, E)$ be a directed acyclic graph (DAG) that represents a partial order $\prec$. Let $w : V \to \mathcal{R}^+$ and $p : V \to \mathcal{R}^+$ be two non-negative weight functions on the nodes. A subset of nodes $S \subseteq V$ is said to be precedence-closed if there is no pair of nodes $(u, v)$ such that $v \in S$, $u \notin S$ and $u \prec v$. Think of the nodes as jobs and the precedence constraints as specifying dependencies; $p(u)$ represents the processing time of job $u$ and $w(u)$ its weight. A natural problem here is to find an ordering (that extends the partial order) of the nodes/jobs to minimize the sum of weighted completion times of the jobs. One can use a precedence-closed set $S$ to decompose the problem into two independent problems on $S$ and $V \setminus S$; recursively find an ordering for $S$, an ordering for $V \setminus S$ and concatenate them. A useful way to decompose the problem is to find a precedence-closed subset $S$ that minimizes the ratio $p(S)/w(S)$ over all precedence-closed subsets. Here $p(S) = \sum_{u \in S} p(u)$ and $w(S) = \sum_{u \in S} w(u)$. Describe a polynomial time algorithm to find such a subset. (*Note:* You are not solving the overall scheduling problem, only the problem of finding a precedence-closed subset of min ratio). *Hint:* Given a number $\lambda$, find an algorithm to decide if there is a precedence-closed set $S$ such that $p(S)/w(S) \leq \lambda$.

**Problem 4:** Let $S$ be a set of $n$ objects (say in the plane). You are given access to a procedure **isCovered**$(S', x)$, such that given a subset $S' \subseteq S$ and a point $x \in \Re^2$, it returns true if $x$ is contained in the union of $S'$ (in other words if there is some object in $S'$ that contains/covers $x$).

(A) Show an efficient deterministic algorithm that takes a point $x$ and returns the number of objects in $S$ that contain $x$. The only direct operation the algorithm can do on $S$, is to call **isCovered**. If $x$ is covered $k$ times by the objects of $S$, how many calls to **isCovered** does your algorithm make? Note that a bound of $n$ is trivial. We are interested in a bound of the form $f(k, n)$ that should be (significantly) smaller than $n$ when $k$ is small (compared to $n$). As a starting point, can you obtain a (significantly) better upper bound than $n$ to distinguish between $k = 1$ and $k > 1$?

(B) Let $R_0 = S$, and let $R_i$ be a random sample from $R_{i-1}$, where we pick every object from $R_{i-1}$ with probability 1/2. Let $H = \langle R_0, \ldots, R_m \rangle$ be the sequence of sets generated in this way (where $R_m$ is the last non-empty set). The sequence $H$ is known as a *gradation* in the literature.

(i) If $x$ is covered $k$ times by objects of $S$, what is the expected number of objects in $R_i$ that cover $x$?

(ii) Let $\alpha_i$ be the probability that $x$ is covered by at least one object of $R_i$. Give a concise expression for $\alpha_i$.

(C) Let $H = \langle R_0, R_1, \ldots \rangle$ and $H' = \langle R'_0, R'_1, \ldots \rangle$ be two gradations generated independently from $S$. Given $x$, let $Y_i$ be the indicator variable that is 1 if $x$ is covered by both $R_i$ and $R'_i$. Let $Z = \sum_i 2^i Y_i$. Prove that $E[Z] = \Theta(\alpha)$ where $\alpha$ is the actual number of objects in $S$ that contain $x$ (you can assume that $\alpha = 2^t$ for some $t \geq 0$ if it helps simplify your arguments). **Comment:** The above procedure easily yields a randomized constant factor approximation algorithm for estimating the number of objects of $S$ covering a query point $x$. In expectation it makes $O(\log n)$ calls to **isCovered**.