Algorithms and Theoretical Computer Science
Ph.D. Qualifying Examination
Fall 2013

Name:	
Alias:	

- Please write your name and an alias of your choice in the boxes above, and submit this page separately from your answers. Please write your alias ***but not your name*** on your answers; this will allow us to grade your exam anonymously.
- The exam consists of eight written questions, four on algorithms (offered on Monday) and four on complexity and formal language theory (offered on Tuesday).
- Please start your answer to each numbered question on a new sheet of paper, so that we can distribute your answers to the faculty who posed each question.
- Write clearly and concisely. You may appeal to standard textbook results (algorithms, data structures, theorems, etc.) unless the problem explicitly asks for details or a proof.
- If you cannot solve a problem completely, you can get partial credit by expressing your main idea/approach clearly and concisely, or by stating necessary lemmas you believe to be true but cannot prove during the exam. All else being equal, it is better to solve some of the problems completely than to get partial credit on every problem.

One of my working assumptions which has been proven successful so often as seemingly to qualify it as a reliable tenet is that "A problem adequately stated is a problem solved theoretically and immediately, and therefore subsequently to be solved, realistically."

— Buckminster Fuller, "Venus Proximity Day" (1965)

A student who changes the course of history is probably taking an exam.

— Unknown

#	Algorithms 1	Algorithms 2	Algorithms 3	Algorithms 4	Complexity 1	Complexity 2	Complexity 3	Complexity 4	Σ
Score									
Max	25	25	25	25	25	25	25	25	200
Grader									

Algorithms and Data Structures

1. A simple path P in an undirected graph G is *induced* if the only edges in G between nodes of P are the edges of P themselves.

   Prove that for all integers $\ell < n$, any tree T with at least n nodes contains either at least ℓ leaves or an induced path of length at least n/ℓ. Conclude that any n-node tree contains either at least $\sqrt{n}$ leaves or an induced path of length at least $\sqrt{n}$.

   Harder problem to think about later: Prove that for some fixed $\delta > 0$, every graph G with n nodes has a spanning tree T that contains either $\Omega(n^\delta)$ leaves or an induced path in G of length $\Omega(n^\delta)$. (In fact, this theorem is true for $\delta = 1/2$.)

2. Consider the sequence $x_0, x_1, x_2, \ldots$, where $x_0 = 1$ and every other element x_i is determined by an independent fair coin flip: with equal probability, either $x_i = x_{i-1}$ or $x_i = x_{i-1}/2$.

   (a) Find the exact value of $\mathrm{E}[x_i]$ for every integer i.

   (b) Prove that there is a constant c such that $\Pr[x_{c \lg n} \geq 1/n] \leq 1/n^{10}$ for every integer n.

   (c) Prove that there is a constant c such that, in a sequence of $c \lg n$ independent fair coin flips, the probability of getting fewer than $\lfloor \lg_2 n \rfloor$ heads is less than $1/n^{10}$.

   (d) Prove that randomized quicksort runs in $O(n \log n)$ time with high probability, where n is the size of the array being sorted.

3. Suppose we had a theorem that states that unless $\mathsf{P} = \mathsf{NP}$, the minimum vertex cover problem has no polynomial-time $(1 + \delta)$-approximation on graphs with maximum degree at most 10, for some fixed constant $\delta > 0$. Using this theorem, prove that unless $\mathsf{P} = \mathsf{NP}$, there is no polynomial-time $(1 - \varepsilon)$-approximation for the maximum clique problem, for some fixed $\varepsilon > 0$.

4. An ***(s, t)*-*series-parallel*** graph is an directed acyclic graph with two designated vertices s (the *source*) and t (the *target* or *sink*) and with one of the following structures:

   - **Base case:** A single directed edge from s to t.
   - **Series:** The union of an (s, u)-series-parallel graph and a (u, t)-series-parallel graph that share a common vertex u but no other vertices or edges.
   - **Parallel:** The union of two smaller (s, t)-series-parallel graphs with the same source s and target t, but with no other vertices or edges in common.

   (a) Describe an efficient algorithm to compute a maximum flow from s to t in an (s, t)-series-parallel graph with arbitrary edge capacities.

   (b) Describe an efficient algorithm to compute a *minimum-cost* maximum flow from s to t in an (s, t)-series-parallel graph whose edges have *unit* capacity and arbitrary costs.

   Harder problem to think about later: How would you efficiently compute a minimum-cost maximum flow in an (s, t)-series-parallel graph with ***arbitrary*** capacities and costs?

Complexity and Formal Languages

1. (a) Prove that for every language $L_1 \subseteq 1^*$, the language L_1^* is regular.

   (b) Prove that there is a language $L_2 \in \{0,1\}^*$ such that L_2^* is not regular.

2. Recall that EXP is the class of languages that can be decided by deterministic Turing machines in $2^{O(n^c)}$ time for some constant $c > 0$, and NEXP is the analogous class for non-deterministic Turing machines. More succinctly,

   $$\mathsf{EXP} = \mathsf{TIME}\big(2^{n^{O(1)}}\big) \quad \text{and} \quad \mathsf{NEXP} = \mathsf{NTIME}\big(2^{n^{O(1)}}\big).$$

   A language L is *unary* if every string in L has the form 1^n for some integer $n \geq 0$.

   Prove that if every unary language in NP belongs to P, then $\mathsf{EXP} = \mathsf{NEXP}$. *[Hint: Be careful not to confuse* EXP *with the class* $\mathsf{E} = \mathsf{TIME}(2^{O(n)})$.*]*

3. BPL the class of languages accepted by logarithmic-space and polynomial-time probabilistic algorithms with error probability at most $1/3$. Such a probabilistic algorithm is modeled as reading a single fresh random bit at every step; old random bits are not available unless recorded on the log-space work-tape.

   Prove that $\mathsf{BPL} \subseteq \mathsf{P}$.

4. Let G be a d-regular n-vertex graph. Let $A(G)$ denote the normalized adjacency matrix of G, defined as $A(G)_{i,j} = (1/d) \cdot \{\#\text{edges between } i \text{ and } j\}$, and let $\lambda = \lambda(G)$ denote the second largest eigenvalue of $A(G)$. We call such a graph G an ***(n,d,λ)-graph***.

   An ***(n,d,ρ)-edge-expander*** is a d-regular n-vertex graph such that for every subset S of vertices with $|S| \leq n/2$, we have

   $$|E(S,\bar{S})| \geq \rho \cdot d|S|$$

   where $\bar{S}$ denotes the complement of S, and $E(S,\bar{S})$ denotes the number of edges between S and $\bar{S}$.

   (a) Prove that if a probability distribution X has support of size at most d, its collision probability is at least $1/d$.

   (b) Fix an (n,d,λ)-graph G, and let X be the distribution over a random neighbor of some vertex. Prove that the collision probability of X is at most $\lambda^2 + 1/n$.

   (c) Prove that $\lambda \geq \sqrt{\frac{1}{d} - \frac{1}{n}} = \frac{1}{\sqrt{d}} + o(1)$.

   (d) Now define an ***(n,d)-random graph*** to be an n-vertex graph G generated by choosing d random permutations $\pi_1, \pi_2, \ldots, \pi_d : [n] \to [n]$ and then letting the edges of G be all pairs $(v, \pi_i(v))$. Prove that a (n,d)-random graph is an $(n, 2d, 1/10)$-edge-expander with probability $1 - o(1)$, that is, with probability approaching 1 as n tends to infinity.