Qualifying Examination
Theoretical Computer Science
Wednesday, March 5, 2014

Part I: Algorithms

Name	

Problem	Maximum Points	Points Earned	Grader
1	25		
2	25		
3	25		
4	25		
5	25		
Total	100		

Instructions:

1. This is a closed book exam.
2. The exam is from 9am–5pm and has five problems. Solve as many as you can but we expect no more than four. The extra problem is to give you some choice. 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: Let T be a rooted tree with root r. Let n be a node in T and let $r = v_0, v_1, v_2, \ldots, v_k = n$ be the unique path from r to n. Then define $w(r) = 1$, and for a node n with parent p, define $w(n) = w(p) \cdot \text{degree}(p)$.

1. Consider the following experiment: Start at the root, and take a random walk to a leaf, at each step selecting a child of a node v uniformly at random (i.e., each with probability $1/\text{degree}(v)$, until a leaf is reached. Let $r = v_0, v_1, v_2, \ldots, v_m$ be the set of vertices visited along this path (v_m is the leaf).

   Define the random variable $t = \sum_{i=0}^{m} w(v_i)$. Prove that $E[t] = |T|$.

2. The first part gives an estimate of the size of a tree by taking a random walk from the root to a leaf, in essence assuming that what is seen along the walk is representative of every other walk in the tree. Give a comparable method to estimate the size of a DAG by taking a random walk from a source to a sink, and prove that the expected value of your estimate is indeed the number of nodes of the DAG. (It may help to initially think about the case in which there is a unique source.)

Problem 2: Describe a polynomial-time algorithm that given an undirected graph $G = (V, E)$ outputs an even-length cycle C in G if it has one, or correctly reports that G does not contain an even-length cycle.

Problem 3: Congratulations! Your research team has just been awarded a $50M multi-year project, jointly funded by DARPA, Google, and McDonalds, to produce DWIM: The first compiler to read programmers minds! Your proposal and your numerous press releases all promise that DWIM will automatically correct errors in any given piece of code, while modifying that code as little as possible. Unfortunately, now its time to start actually making the damn thing work.

As a warmup exercise, you decide to tackle the following necessary subproblem. Recall that the *edit distance* between two strings is the minimum number of single-character insertions, deletions, and replacements required to transform one string into the other. An *arithmetic expression* is a string w such that

- w is a string of one or more decimal digits,
- $w = (x)$ for some arithmetic expression x, or

- $w = x \diamond y$ for some arithmetic expressions x and y and some binary operator $\diamond$.

Suppose you are given a string of tokens from the alphabet $\{\#, \diamond, (,)\}$, where # represents a decimal digit and $\diamond$ represents a binary operator. Describe an algorithm to compute the minimum edit distance from the given string to an arithmetic expression.

Problem 4: Let G be an n vertex d-regular graph.

1. We say that a distribution p over the vertices of G (where p_i denotes the probability that vertex i is picked by p) is stable if when we choose a vertex i according to p and take a random step from i (i.e., move to a random neighbor j of i) then the resulting distribution is p. Prove that the uniform distribution on the vertices of G is stable.

2. For p a distribution over the vertices of G, let $\Delta(p) = \max_i\{p_i - 1/n\}$. For every k, denote by p^k the distribution obtained by choosing a vertex i at random from p and taking k random steps on G. Prove that if G is connected then there exists k such that $\Delta(p^k) \leq (1 - n^{10n})\Delta(p)$. Conclude that

   - The uniform distribution is the only stable distribution for G.
   - For every vertices u, v of G, if we take a sufficiently long random walk starting from u, then with high probability the fraction of times we hit the vertex v is roughly $1/n$. That is, for every $\epsilon > 0$, there exists k such that the k-step random walk from u hits v between $(1-\epsilon)k/n$ and $(1+\epsilon)k/n$ times with probability at least $1-\epsilon$.

3. For a vertex u in G, denote by E_u the expected number of steps it takes for a random walk starting from u to reach back u. Show that $E_u \leq 10n^2$.

4. For every two vertices u,v denote by $E_{u,v}$ the expected number of steps it takes for a random walk starting from u to reach v. Show that if u and v are connected by a path of length at most k then $E_{u,v} \leq 100kn^2$. Conclude that for every s and t that are connected in a graph G, the probability that an $1000n^3$ random walk from s does not hit t is at most 1/10.

5. Let G be an n-vertex graph that is not necessarily regular (i.e., each vertex may have different degree). Let G' be the graph obtained by adding a sufficient number of parallel self-loops to each vertex to make G regular. Prove that if a k-step random walk in G from a vertex s hits a vertex t with probability at least 0.9, then a $10n^2k$-step random walk from s will hit t with probability at least 1/2.

Problem 5: An edge cover in an undirected loop-less graph $G = (V, E)$ is a subset of edges $E' \subseteq E$ such that each node is incident to some edge in E'. In vertex cover we cover edges by nodes while in edge cover we cover nodes by edges. Unlike vertex cover, egde cover is polynomial-time solvable due its connections to matchings. Show that in any graph G we have $\rho(G) + \nu(G) = |V|$ were $\rho(G)$ is the cardinality of a minimum edge-cover in G and $\nu(G)$ is the cardinality of a maximum matching in G.