Qualifying Examination

Theoretical Computer Science

Wednesday, March 16, 2015

Part I: Algorithms

Name	

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 from 10:30am–6:30pm 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 matroid $\mathcal{M}$ is a tuple $(N, \mathcal{I})$ where N is finite ground set and $\mathcal{I} \subseteq 2^N$ is a collection of *independent sets*, that satisfy the following conditions.

1. $\mathcal{I}$ is non-empty, in particular, $\emptyset \in \mathcal{I}$.
2. $\mathcal{I}$ is downward closed, that is, if $A \in \mathcal{I}$ and $B \subset A$ then $B \in \mathcal{I}$.
3. If $A, B \in \mathcal{I}$ and $|A| < |B|$ then there is an element $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

Let $G = (V, E)$ be a directed graph and let $r \in V$ and $T \subseteq V \setminus \{r\}$ be a set of terminals. We say that $T' \subseteq T$ is routable to r if there is a collection of edge-disjoint paths connecting T' to r with one path for each $t \in T'$. Consider $\mathcal{M} = (T, \mathcal{I})$ where $\mathcal{I} = \{T' \subseteq T \mid T' \text{ is routable to } r \text{ in } G\}$. Show that $\mathcal{M}$ is a matroid.

Problem 2: Recall Karger's randomized algorithm for the global minimum cut problem. The algorithm starts with a multigraph $G = (V, E)$ and picks an edge $uv \in E$ uniformly at random and contracts the end points u, v. It repeats this procedure until the graph has exactly two nodes left and outputs the induced partition as the mincut. Recall that this algorithm is shown to output a correct global mincut with probability $\Omega(1/n^2)$. Now consider the k-cut problem where the goal is to partition the graph into k non-empty vertex subsets $V_1, V_2, \ldots, V_k$ so as to minmize the number of edges crossing the partition. Generalize the global mincut's analysis to obtain a randomized Las Vegas algorithm that outputs the min k-cut in a given graph with probability at least $1/2$. What is the running time of your algorithm as a function of n and k.

Problem 3: Suppose you are given n nuts and n bolts of different but nearly identical sizes. You can easily determine whether any nut is smaller or larger than any bolt, but you cannot directly compare two nuts or two bolts. **Unlike the standard nuts and bolts problem, there are no matching nut-bolt pairs.** Instead, the nuts and bolts alternate in size. That is, for any pair of bolts, at least one nut is larger than one and smaller than the other, and for any pair of nuts, at least one bolt is larger than one and smaller than the other.

Describe and analyze a randomized algorithm to sort the nuts and bolts by size in $O(n \log n)$ expected time.

Problem 4: Given a set P of n points in general position, a triple of points $p, q, r \in P$ is *Delaunay*, if the circle passing through p, q, r contains no points of P

in its interior. Let $D_0(P)$ be this set of triplets. It is known that there are $O(n)$ such triplets, and they can be computed in $O(n \log n)$ time, using a procedure called `computeDelaunayTriangulation`. A triple $p, q, r \in P$ is *$\leq k$-Delaunay* if the circle passing through p, q, r contains at most k points of P. Let $D_{\leq k}$ be the set of such *$\leq k$-Delaunay* triplets.

1. Prove, that $|D_{\leq k}| = O(k^2 n)$ (hint: think about a random sample R of P, where every point is picked with probability p, and what happens to a triplet of $D_{\leq k}$. Specifically, consider the probability that it is in $D_0(R)$. Also, consider the expected value of $|D_0(R)|$. (You can safely assume that, with high probability. we have $pn/2 \leq |R| \leq 2pn$.)

2. Provide a randomized algorithm that computes all the triplets of $D_{\leq k}$ in $O(nk^3 \log^{O(1)} n)$ time. The algorithm should succeed with high probability. (Hint: Use (A) and `computeDelaunayTriangulation`.)