PhD Qualifi Exam for Spring 2007–Theory Area

Spring 2007, CS Department, IIT
Your random number___________1

There are 6 questions in this exam. For every question, please write your answer in a clean and concise way. If you are asked to write an algorithm for a question, you have to write the pseudo-code of your algorithm and also put explanation about your pseudo-code.

1. Let $G = (V,E;r)$ be a weighted connected graph. For any edge $e \in E$, the value $r(e)$ is referred to as the reliability of $e$. For any path $P$ in $G$, the reliability of $P$ is, by definition, the minimum reliability of the edges occurring in $P$. The reliability $r_G(s,t)$ of two vertices $s$ and $t$ is equal to the maximum reliability of $P$ where $P$ ranges over all $s-t$ paths.

(a) Give an algorithm with running time $O(m + n \log n)$ to compute $r_G(s,t)$ for any $s,t \in V$.

(b) Prove that if $T$ is a longest spanning tree (i.e., $\sum_{e \in E(T)} r(e)$ is the largest among all spanning trees), then $r_T(s,t) = r_G(s,t)$ for all $s,t \in V$.

2. Let $G = (V,E)$ be a connected graph, and $T$ be a depth-first spanning tree (DFS tree) of $G$. For each $v \in V$, denote by $T(v)$ the sub-tree of $T$ induced by the descendants of $v$ (including $v$ itself), and by $G - v$ the subgraph of $G$ induced by $V \setminus \{v\}$. Prove that $G$ is biconnected (i.e., for each node $v \in V$, the graph $G - v$ is connected) if and only if the root of $T$ has exactly one child and for each node $v$ other than the root and its (unique) child, there is an edge between a node in the subtree $T(v)$ and a proper ancestor of $v$ other than the parent of $v$. (Hint: you may use the fact that each edge of $G$ is between a vertex and one of its descendants.)

3. Let $G = (V,E)$ be an undirected graph with $|V| = n$, and $s$ be a fixed node in $G$. The depth of a node $v$ is the distance between $v$ and $s$. Denote by $R$ the maximum distance of all the nodes from $s$. For $0 \leq i \leq R$, the layer $i$ of $G$ consists of all nodes of depth $i$. The following procedure computes a special BFS tree $T$ referred to as canonical BFS tree and an associated ranking $\text{rank}$ of the nodes constructed layer-by-layer in the bottom-up manner. Initially, $T$ is empty and $\text{rank}(v) = 0$ for each node $v$ at the layer $R$. The ranks and the children of all nodes at each other layer $i$ are computed iteratively: Initialize $U$ to be the set of nodes at layer $i$, and $W$ to be the set of nodes at layer $i + 1$. Repeat the following iteration while $W$ is non-empty. Compute the maximum rank $r$ of the nodes in $W$, and find a node $v \in U$ which is adjacent to the largest number of nodes in $W$ with rank $r$. If $v$ is adjacent to only one node in $W$ with rank $r$, then $\text{rank}(v) = r$; otherwise, $\text{rank}(v) = r + 1$. Put all neighbors of $v$ in $W$ as the children of $v$ in $T$. Remove $v$ from $U$, and remove all neighbors of $v$ from $W$. When $W$ is empty, set $\text{rank}(v) = 0$ for each node $v \in U$. Figure 1 gives an example of the ranking and the canonical BFS tree constructed in this way. Prove that (1) for each $v \in V$, $\text{rank}(v) \leq \lfloor \log n \rfloor$; and (2) if $u_1$ and $u_2$ are two nodes at the same layer, $v_1$ and $v_2$ are their child respectively at layer $i + 1$, and all of them have the same rank, then neither $u_1$ and $v_2$ nor $u_2$ and $v_1$ are adjacent in $G$. 

4. Solve problems 1, 5, 6, and any one out of 2, 3, 4.
4. Let $D = (V, A; c)$ be a weighted digraph. A cycle cover of a graph $D$ is a collection of vertex-disjoint cycles such that every vertex of $G$ is a part of a cycle. The weight of a cycle cover is the total weight of edges in this cycle cover. Construct a weighted bipartite $H = (X \cup Y, E; c')$ as follows: Let $V = \{v_1, v_2, \ldots, v_n\}$. Then $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ are two disjoint copies $V$. For each edge $v_i v_j \in A$, add an edge $x_i y_j$ to $E$ with weight $c'(x_i y_j) = c(v_i v_j)$. Prove that $D$ contains a cycle cover of weight $C$ if and only if $H$ contains a perfect matching of weight $C$. (Recall that a perfect matching in $H$ is a set of $n$ node-disjoint edges.)

5. Suppose $S$ is a sequence of numbers divided into $m$ consecutive sub-sequences $S_1, S_2, \ldots, S_m$, each of which is sorted. In order to sort $S$, we may and are only allowed to merge adjacent sub-sequences into a larger sorted sub-sequence. The the number of comparisons of merging two sorted sub-sequences into a larger sorted sub-sequence is the total length of the two sub-sequences minus one. Give a dynamic programming algorithm to find the best order to combine the sub-sequences so as to have the smallest total number of comparisons.

6. Assume that you only know the following problems are NP-complete: SAT/CNF-SAT, 3-SAT, Vertex Cover, Clique, Independent Set, Hamiltonian Path/Cycle, Subset Sum. Consider the following MINIMUM BIN PACKING problem. Given a finite $U$ of items with a size $s(u) \in \mathbb{Z}^+$ for each $u \in U$ and a positive integer bin capacity $B$, seek a partition of $U$ into the smallest number of disjoint sets $U_1, U_2, \ldots, U_m$ satisfying that $\sum_{u \in U_j} s(u) \leq B$ for each $1 \leq j \leq m$. Prove that the decision version of MINIMUM BIN PACKING is NP-complete.