Data Structures and Algorithms (II)

close-book midterm exam

April 30, 2007

**Instruction** You may answer the questions in any order. Dishonest behaviors and attempts will be most seriously punished. When you are asked to justify, prove, or disprove your answers, you may directly use anything that we have shown in class in a “black-box” manner unless the things you are asked to do have been shown in class.

**Problem 1** (10 points)

State the first two cases of the master method. That is, you do not have to worry about the case which is something like \( f(n) = \Omega(n^{\log_b a + \epsilon}) \).

**Problem 2** (20 points)

We showed in class how to solve the bipartite matching problem in polynomial time by reducing it to a maximum flow problem. In particular, if \( G = (V_L, V_R, E) \) is an undirected bipartite graph, we construct a unit-capacity directed graph \( G^* \) from \( G \) by

- orienting all edges of \( G \) from \( V_L \) to \( V_R \);
- adding a source node \( s \) that has an outgoing edge to each node in \( V_L \); and
- adding a sink node \( t \) that has an incoming edge from each node in \( V_R \).

We proved in class the statement that “\( G \) has a maximum matching with \( k \) edges if and only if \( G^* \) has a maximum flow with value \( k \).” Now you are asked to prove or disprove the following slightly different statement for any integer \( k \):

\[ G \text{ has a matching with } k \text{ edges if and only if } G^* \text{ has a flow with value } k. \]

**Problem 3** (20 points)

We sketched the proof of Edmonds-Karp’s theorem saying that the number of augmentation-through-shortest-path phases is \( O(nm) \), where \( n \) is the number of nodes and \( m \) is the number of edges in the input graph for the maximum flow problem. You are now asked to prove the theorem as rigorously as possible.
Problem 4  (15 points)

We mentioned that it takes $O(mn)$ time to determine whether an $n$-node $m$-edge directed graph $G$ has negative cycles. We said in class that it can be done by running Bellman-Ford from an arbitrarily chosen root $r$ for $n$ phases of estimate improvement. We claimed that

$G$ has negative cycles if and only if the estimate of distance from $r$ still changes in the $n$-th phase of estimate improvement.

Unfortunately, the above claim is wrong when the negative cycles of $G$ cannot be reached from the arbitrarily chosen root $r$. For example, consider the case that $G$ contains a single negative cycle and the negative cycle is not reachable from $r$. That is, the nodes on the negative cycle have infinite distance from $r$ in $G$.

One way to fix the algorithm is to running Bellman-Ford for all $n$ roots, but that would require $\Theta(mn^2)$ time.

Now you are asked to give an algorithm that correctly determines whether the input graph has negative cycles in $O(mn)$ time. Justify your answer.

Problem 5  (15 points)

Let $G$ be a graph whose edge weights are distinct. Let $T$ be a spanning tree of graph $G$. An edge $(u, v)$ of $G - T$ is $T$-heavy if $(u, v)$ is heavier than any edge on the unique path of $T$ connecting $u$ and $v$. You are asked to prove the following heaviness lemma

$T$ is the minimum spanning tree of $G$ if and only if each edge of $G - T$ is $T$-heavy.

Problem 6  (20 points)

Let $S$ be a subset of the nodes of graph $G$. We say that $S$ is an independent set of $G$ if any two nodes of $S$ are not adjacent in $G$.

A ladder graph is a graph consisting of two paths of equal length $n$ where the $i$-th node of each path is connected to each other by an edge. Suppose that the weight of each node in the graph is bounded by a constant. What is the worst-case time complexity of the problem of finding a maximum-weight independent set for an $2n$-node ladder graph? Justify your answer.