Assignment 9. Graphs

Given April 18, due April 25.

1. We have a connected undirected graph, $G$, with all edges marked either red or blue (but not both). Let $G_b$ be the subgraph having the same vertices but only the blue edges. The connected components of $G_b$ are $C_1, \ldots, C_l$. The “reduced graph” has one vertex for each of the $C_k$ and an edge from $C_k$ to $C_j$ if there is a red edge in $G$ that goes from a vertex in $C_k$ to a vertex in $C_j$.

a. Suppose that $C_k$ is connected to $C_j$ in the above sense, and that $v \in C_k$, and $u \in C_j$ are two vertices of $G$. Show that there is a path in $G$ from $v$ to $u$ that visits only vertices from $C_k$ and $C_j$.

b. Show that, conversely, if there is a path from $v$ to $u$ as in part a., then the component containing $v$ and the component containing $u$ are connected in the reduced graph.

c. Sketch an algorithm that constructs the reduced graph (makes an array of vertices and edge lists for the vertices), using the set operations makeset, find, and union.

2. In a directed graph, the indegree of a vertex $x$ is the number of edges whose head is $x$. The outdegree is the number of vertices whose tail is $x$. Suppose $G$ is a directed graph given by an array of $v$ vertices and, for each vertex $x \in V$, a linked list of other vertices in $V$. If $y$ is in this linked list, then there is an edge whose tail is $x$ and head is $y$. To find the outdegree of $x$ we simply count the number of elements in the linked list at $x$. Give pseudocode that describes an algorithm that determines the indegree of each vertex. The work for this should be on the order of $e + v$.

3. For a directed graph described using the above data structure (a linked list of vertices to represent edges), give an algorithm related to breadth first search that creates a list, of all vertices that can be reached from $x$ using no more than $R$ hops. Call this list $W$. The work for this algorithm should be proportional to the number of vertices in $W$ plus the number of edges whose tail is in $W$. Warning: This is a little subtle. The first step of normal breadth first search is to mark each vertex in $V$ as “white” (CLR terminology) or “unvisited” (more common term). If $V$ is much larger than $W$, this is not allowed. You must find a different way to determine whether a given vertex has been visited or not.