Problem Set 8—Due Friday, November 21

Problem 30 A graph $G = (V, E)$ is said to be ***bipartite*** iff we can partition V into V_1 and V_2 such that for all $(u, v) \in E$ either $u \in V_1$ and $v \in V_2$ or $v \in V_1$ and $u \in V_2$. Recall that to partition V into V_1 and V_2 means that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$.

Design an algorithm to determine whether a given G is bipartite. Justify the correctness and running time of your algorithm.

Problem 31 Given a connected undirected graph G, find a vertex whose removal does not disconnect G. *(Hint: Using DFS or BFS on the graph will yield useful information!)*

Prove your algorithm correct and analyze its running time. Note that proving your algorithm correct will indirectly yield a proof that such a vertex always exists (not immediately obvious).

Problem 32 Text Problem 7-1 (page 152)

Problem 33 Let $G = (V, E)$ be an n-node weighted digraph, $V = \{1, \ldots, n\}$, with edge weights specified by a matrix $W = (w_{i,j})$, $w_{i,j} \in \mathbf{R}^+$. Suppose you have already computed a distance matrix $D = (d_{i,j})$, where each $d_{i,j}$ is the weight of a shortest $i \to j$ path.

(a) Give an $O(1)$ algorithm which, on input i, j, k, ℓ, finds the length of a shortest $i \to j$ path which passes through vertices k and ℓ.

(b) Give an $O(n)$ algorithm which, on input i and j, lists all the vertices v such that v resides on *some* shortest $i \to j$ path.

(c) Give an $O(n \lg n)$ algorithm which, on input i and j, outputs, in order, the vertices of some shortest $i \to j$ path.

Problem 34 Text Problem 25-2 (page 546)