



Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected activities. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.

Suppose that we are given a weighted, directed graph  $G = (V, E)$  in which edges that leave the source vertex  $s$  may have negative weights, all other edge weights are nonnegative, and there are no negative-weight cycles. Argue that Dijkstra's algorithm correctly finds shortest paths from  $s$  in this graph.

Most graph algorithms that take an adjacency-matrix representation as input require time  $\Omega(V^2)$ , but there are some exceptions. Show how to determine whether a directed graph  $G$  contains a *universal sink*—a vertex with in-degree  $|V| - 1$  and out-degree 0—in time  $O(V)$ , given an adjacency matrix for  $G$ .





21. Let us say that a graph  $G = (V, E)$  is a *near-tree* if it is connected and has at most  $n + 8$  edges, where  $n = |V|$ . Give an algorithm with running time  $O(n)$  that takes a near-tree  $G$  with costs on its edges, and returns a minimum spanning tree of  $G$ . You may assume that all the edge costs are distinct.
22. Consider the Minimum Spanning Tree Problem on an undirected graph  $G = (V, E)$ , with a cost  $c_e \geq 0$  on each edge, where the costs may not all be different. If the costs are not all distinct, there can in general be many distinct minimum-cost solutions. Suppose we are given a spanning tree  $T \subseteq E$  with the guarantee that for every  $e \in T$ ,  $e$  belongs to *some* minimum-cost spanning tree in  $G$ . Can we conclude that  $T$  itself must be a minimum-cost spanning tree in  $G$ ? Give a proof or a counterexample with explanation.

