1. [Goodrich & Tamassia, C-7.3] Consider a greedy strategy for finding a path from vertex \textit{start} to vertex \textit{goal} in a given connected graph, that performs the following steps:

(a) $\text{path} = [\text{start}]$

(b) $\text{visited} = \{\text{start}\}$

(c) if $\text{start} == \text{goal}$, return $\text{path}$ and exit

(d) find the minimum-weight edge $\text{start} \rightarrow \text{next}$ such that $\text{next}$ is not in $\text{visited}$

(e) add $\text{next}$ to the end of $\text{path}$

(f) add $\text{next}$ to $\text{visited}$

(g) set $\text{start}$ to $\text{next}$ and go to step (c)

Does this greedy strategy always find a shortest path from $\text{start}$ to $\text{goal}$? Either explain intuitively why it works, or give a counterexample.

2. [Goodrich & Tamassia, C-7.6] Design an efficient algorithm for finding a \textit{longest} directed path from a vertex $s$ to a vertex $t$ of a weighted directed acyclic graph $G$. Also, analyze the time complexity of your algorithm.

3. Find a directed acyclic graph $G$, with a starting vertex $s$ that can reach all other vertices of $G$, such that if you use Dijkstra’s algorithm to find shortest paths in $G$ from $s$, the order in which it finds these paths is \textit{not} a topological ordering of $G$.

4. Suppose we are given a directed graph $G$ with positive edge weights, and vertices $s$ and $t$ in the graph. We wish to find the path from $s$ to $t$ that minimizes the \textit{product} of the weights of its edges (rather than minimizing the sum of the weights as in the usual shortest path problem).

(a) Describe a method for computing a new set of weights for the edges of $G$, such that the shortest path from $s$ to $t$ for the new weights is automatically the same as the minimum-product path for the original weights.

(b) Under what conditions on the original edge weights will Dijkstra’s algorithm (using the new weights) be guaranteed to find the correct path?