Minimum Spanning Trees

1. After calculating the minimum spanning tree, you decide to add a positive constant to all edge weights within the graph. Will this operation change the minimum spanning tree of the graph?

    ....

    ....

    ....

    ....

2. A colleague suggests the following algorithm to find the minimum spanning tree: Start with any vertex as a single-vertex minimum spanning tree, then add $V - 1$ edges to it, always taking next a min-weight edge connected to the vertex most recently added. Why does your colleague's algorithm not necessarily always produce a minimum spanning tree?

    ....

    ....

    ....

    ....

3. A colleague argues that every graph has only one possible minimum spanning tree? Is your colleague correct? Under what cases can a graph have multiple minimum spanning trees?

    ....

    ....

    ....

    ....

4. Describe an algorithm on how to find a *maximum* spanning tree?

    ....

    ....

    ....

    ....

5. Suppose that a graph has distinct edge weights. Does the edge with the smallest weight have to belong to the minimum spanning tree? Can the edge with the largest weight belong to the minimum spanning tree? Why or why not?

    ....

    ....

    ....

    ....

Shortest Paths

6. After calculating the shortest paths from a single source to all other vertices in a graph, you decide to add a positive constant to all edge weights within the graph. Will this operation change the shortest paths of the graph?

7. Why doesn't Dijkstra's algorithm work for a graph with negative edge weights?

8. A colleague thinks that a minimum spanning tree algorithm can also be used to determine the shortest paths between vertices in a graph. Is your colleague correct? Why or why not?

9. How can you modify Dijkstra's algorithm to efficiently find the longest paths between a single source vertex and all other vertices? Assume there are no cycles within the graph.

10. You can use Dijkstra's algorithm to find the shortest path from a source vertex to all other vertices in the graph in $O(E + V \log_2 V)$ time. However, suppose you have a directed graph where each edge has a weight of either 1 or 2. For such a graph, how can you find the shortest path from a source vertex to all other vertices in the graph in $O(V + E)$ time?

11. Suppose you want to calculate the shortest path from some n vertices to all other vertices in the graph. It might be tempting to run Dijkstra's algorithm n times. However, there is a more efficient way. How can you modify either the graph or Dijkstra's algorithm to support multiple source vertices?