CS21003 - Tutorial 9

October 27th, 2018

1. Suppose that you are given a graph $G = (V, E)$ and its minimum spanning tree T. Suppose that we delete from G, one of the edges $(u, v) \in T$ and let G' denotes this new graph.
   (a) Is G' guaranteed to have a minimum spanning tree?
   (b) Assuming that G' has a minimum spanning tree T'. TRUE or FALSE: the number of edges in T' is no greater than the number of edges in T. Explain your answer in one sentence.
   (c) Assuming that G' has a minimum spanning tree T', describe an algorithm for finding T'. What is the complexity of your algorithm?
2. Let G be an undirected connected weighted graph. Suppose the graph has at least one cycle (choose one). For that chosen cycle, let edge e be an edge that has strictly greater cost than all other edges in the cycle. (Such an edge might not exist, e.g., there might be two edges that have the same greatest cost). Show that e does not belong to any MST of G.
3. Consider a "reversed" Kruskals algorithm for computing an MST. Initialize T to be the set of all edges in the graph. Now, consider edges from largest to smallest cost. For each edge, delete it from T if that edge belongs to a cycle in T. Assuming all the edge costs are distinct, does this new algorithm correctly compute an MST?
4. A *maximum spanning tree* of G is a spanning tree T of G such that the sum of costs of the edges in T is as large as possible. Modify Kruskal's algorithm to compute a maximum spanning tree of G. What is the running time of your algorithm?
5. We are given a directed graph $G(V, E)$ on which each edge (u, v) has an associated value $r(u, v)$, which is a real number in the range $[0, 1]$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.
6. Let $G = (V, E)$ be a weighted graph and T be the array containing the parents (previous vertices) corresponding to the shortest distance from source s to all the other vertices. Assume all weights in G are increased by the same amount c. Is T still the parent array (from source s) of the modified graph? If yes, prove the statement. Otherwise, give a counter example.
7. Let $G = (V, E)$ be a weighted undirected graph. Let $s, t \in V$ and $s \neq t$. Design an $O(ElogV)$ algorithm to find all vertices v such that v lies on at least one of the shortest paths between s and t.
8. Given a directed graph $G = (V, E)$ with vertex set $V = \{1, 2, \ldots, n\}$, we define the *transitive closure* of G as the graph $G^* = (V, E^*)$, where
   $$E^* = \{(i, j) : \text{there is a path from vertex } i \text{ to vertex } j \text{ in } G\}$$
   Can you propose an $O(n^3)$ algorithm to compute the transitive closure of the graph?
9. Describe an algorithm to find the length of the shortest cycle in a weighted directed graph in $O(n^3)$ time. Assume that all the edge weights are positive.

10. Let D be the shortest path matrix of an undirected weighted graph G. Thus $D(u, v)$ is the length of the shortest path from vertex u to vertex v, for every two vertices u and v. Graph G and matrix D are given. Assume the weight of an edge $e = (a, b)$ is decreased from w_e to w'_e. Design an algorithm to update matrix D with respect to this change. The running time of your algorithm should be $O(n^2)$. Describe all details and write a pseudo-code for your algorithm.