CSC373 Worksheet 4

August 3, 2020

Source: link

1. **CLRS 22.1-1:** Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of every vertex? How long does it take to compute the in-degrees?

2. **CLRS 22.1-3:** The transpose of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$, where $E^T = \{(v, u) \in V \times V : (u, v) \in E\}$. Thus, G^T is G with all its edges reversed. Describe efficient algorithms for computing G^T from G, for both the adjacencylist and adjacency-matrix representations of G. Analyze the running times of your algorithms.

3. **CLRS 22.2-8:** The diameter of a tree $TD.V; E/$ is defined as maxu;2V ı.u; /, that is, the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyze the running time of your algorithm.

4. **CLRS 22.3-2:** Show how depth-first search works on the graph of Figure 22.6. Assume that the for loop of lines 5–7 of the DFS procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and show the classification of each edge.

5. **CLRS 23.1-1:** Let (u, v) be a minimum-weight edge in a connected graph G. Show that (u, v) belongs to some minimum spanning tree of G.

6. **CLRS 23.2-2:** Suppose that we represent the graph $G = (V, E)$ as an adjacency matrix. Give a simple implementation of Prim's algorithm for this case that runs in $\mathcal{O}(V^2)$ time.

7. **CLRS 24.1-3:** Given a weighted, directed graph $G = (V, E)$ with no negative-weight cycles, let m be the maximum over all vertices $v \in V$ of the minimum number of edges in a shortest path from the source s to v. (Here, the shortest path is by weight, not the number of edges.) Suggest a simple change to the Bellman-Ford algorithm that allows it to terminate in $m + 1$ passes, even if m is not known in advance.

8. **CLRS 24.1-4:** Modify the Bellman-Ford algorithm so that it sets :d to 1 for all vertices for which there is a negative-weight cycle on some path from the source to .