CS 170 DIS 04

Released on 2019-02-11

1 Short Answer

For each of the following, either prove the statement is true or give a counterexample to show it is false.

(a) If (u, v) is an edge in an undirected graph and during DFS, $\text{post}(v) < \text{post}(u)$, then u is an ancestor of v in the DFS tree.

(b) In a directed graph, if there is a path from u to v and $\text{pre}(u) < \text{pre}(v)$ then u is an ancestor of v in the DFS tree.

(c) In any connected undirected graph G there is a vertex whose removal leaves G connected.

2 Midterm Prep: Dijkstra Tiebreaking

We are given a directed graph G with positive weights on its edges. We wish to find a shortest path from s to t, and, among all shortest paths, we want the one in which the longest edge is as short as possible. How would you modify Dijkstra's algorithm to this end?

3 Dijkstra's Algorithm Fails on Negative Edges

Draw a graph with five vertices or fewer, and indicate the source where Dijkstra's algorithm will be started from.

1. Draw a graph with no negative cycles for which Dijkstra's algorithm produces the wrong answer.

2. Draw a graph with at least two negative weight edge for which Dijkstra's algorithm produces the correct answer.

4 Midterm Prep: Graph Short Answer

Answer each question below *concisely* (one short sentence or a number should suffice). Do not justify your answer. Do not show your work.

(a) Suppose we are given a directed graph $G = (V, E)$ represented in adjacency list format, and we want to test whether G is a dag or not, using a method that is as asymptotically efficient as possible. In a sentence, what approach would you use?

(b) What's the running time of your solution in (a), using $O(\cdot)$ notation?

(c) Let $G = (V, E)$ be a directed graph with $|V| = 1000$ vertices, $|E| = 5000$ edges, and 700 strongly connected components. How many vertices does the metagraph have?

(d) Let $G = (V, E)$ be a dag. Let s be a source vertex in G. Suppose we set the weight of each edge to 1 and run Dijkstra's algorithm to compute the distance from s to each vertex $v \in V$, and then order the vertices in increasing order of their distance from s. Are we guaranteed that this is a valid topological sort of G?

Circle YES or NO.

(e) Justify your answer to part (d) as follows: If you circled YES, then give one sentence that explains the main idea in a proof of this fact. If you circled NO, then give a small counterexample (a graph with at most 4 vertices) that disproves it.

(f) Suppose we run Dijkstra's algorithm on a graph with n vertices and $O(n \lg n)$ edges. Assume the graph is represented in adjacency list representation. What's the asymptotic running time of Dijkstra's algorithm, in this case, if we use a binary heap for our priority queue? Express your answer as a function of n, and use $O(\cdot)$ notation.

5 The Greatest Roads in America

Arguably, one of the best things to do in America is take a great American road trip. And in America there are some amazing roads to drive on (think Pacific Crest Highway, Route 66, etc). An intrepid traveller has chosen to set course across America in search of some amazing driving. What is the length of the shortest path that hits at least k of these amazing roads?

Assume, that the roads in America can be expressed as a directed weighted graph $G = (V, E, d)$ and that our traveller wishes to drive across at least k roads from the subset $R \subset E$ of 'amazing' roads. Furthmore, assume that the traveller starts and ends at her home $h \in V$. Also, you can assume that the traveller is OK with repeating roads from R i.e. the k roads she chooses from R do not need to be unique.

Provide a 4 part solution with runtime in terms of $n = |V|, m = |E|, k$, and $r = |R|$.

Hint: First try out $k = 1$. How can G be modified so that we can use a 'common' algorithm to solve the problem?