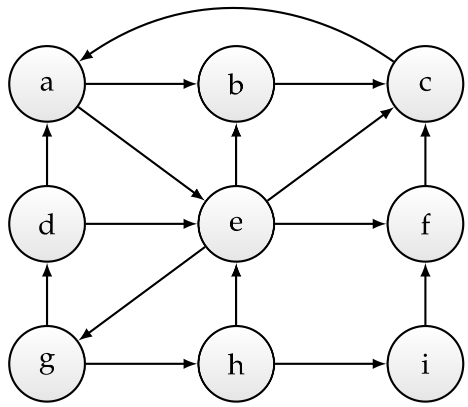
CS3000: Algorithms & Data — Fall '18 — Jonathan Ullman

Homework 6
Due Friday November 2 at 11:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday November 2 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.

Problem 1. *DFS and Topological Ordering*



Consider running depth-first search on this graph starting from node a. When there are multiple choices for the next node to visit, go in alphabetical order.

(a) Label every edge as either tree, forward, backward, or cross.

Solution:

(b) Give the post-order numbers of all vertices

Solution:

(c) Is this graph a DAG? Support your answer by either showing a topological ordering or a directed cycle.

Solution:

Problem 2. *Cleaning up the Streets*

You have been hired to clean up the streets of Boston using your street sweeper. In order to do the job for the lowest cost, you want to devise a way to sweep each street exactly one time in each direction. That is, for each street, you go up the street once and down the street once to sweep each side of the road. Assume all roads are two-way streets.

Specifically, your city is modeled as a graph $G = (V, E)$, where the vertices represent the intersections and edges represent the roads connecting intersections. Design an algorithm based on depth-first search that takes the graph $G = (V, E)$ and finds a street-sweeping route that goes along each edge exactly once in each direction. Your algorithm should run in $O(V + E)$ time.

(a) Explain in a few English sentences how your algorithm works.

Solution:

(b) Describe your algorithm in pseudocode.

Solution:

(c) Justify that your algorithm finds a correct street-sweeping route. Your justification may take any form you like, as long as it is clear and convincing.

Solution:

(d) Analyze the running time of your algorithm.

Problem 3. *All-Pairs Shortest Paths*

In the *all-pairs shortest paths* problem, you are given a directed, weighted graph with edge lengths $G = (V, E, \{\ell_e\})$, and have to find the length of the shortest path from s to t for *every pair* $s, t \in V$. For this HW we only want the *length* of the shortest path and not the path itself.

If all edge lengths are non-negative ($\ell_e \geq 0$), then we can solve this problem by running Dijkstra's algorithm from every source node $s \in V$, incurring running time $O(nm \log n)$. However, if lengths can be negative, then running Bellman-Ford from each source node $s \in V$ incurs running time $O(n^2 m)$. In this question we will study the following algorithm for solving all-pairs shortest paths in graphs with negative-length edges, but no negative-length cycles.

- Modify the input graph by adding an additional node z connected to every other node v by a zero-length edge (z, v).
- Run the Bellman-Ford algorithm on the modified graph with source z to find the length $f(v)$ of the shortest $z \to v$ path in the modified graph.
- Define new edge lengths $\ell'_{u,v} = \ell_{u,v} + f(u) - f(v)$ and let $G' = (V, E, \{\ell'_e\})$ be the input graph with these modified edge weights.
- For each source $s \in V$, run Dijkstra's algorithm on the graph G' with source s to find the length $d'(s, v)$ of the shortest $s \to v$ path in G' for every node v.
- For every $u, v \in V$, let $d(u, v) = d'(u, v) - f(u) + f(v)$. Output the values $\{d(u, v)\}$.

In this problem, we will show correctness and analyze the running time of this algorithm. The final three steps of the problem form the proof of correctness.

(a) What is the running time of this algorithm? Briefly explain your answer.

Solution:

(b) Prove that every edge in G' has non-negative length. That is, $\forall u, v \in V, \ell'_{u,v} \geq 0$. (Thus, Dijkstra's algorithm will correctly find the length $d'(u, v)$ of the shortest $u \to v$ path in G'.)

Solution:

(c) Prove that for every $u \to v$ path $P = u \to w_1 \to \cdots \to w_{k-1} \to v$, we have $\ell'_P = \ell_P + f(u) - f(v)$ where ℓ'_P, ℓ_P are the length of the path in G' and G, respectively.

Solution:

(d) Prove that for every $u, v \in V$, $d'(u, v) - f(u) + f(v)$ is the length of the shortest $u \to v$ path in the original graph G. Thus, the final lengths output by this algorithm are correct.

Solution: