CS4800 - ALGORITHMS - S'17 — NGUYỄN
DUE FRI APR 7, 2017 AT 11.59A VIA GRADESCOPE

H9.

- The assignment is due at Gradescope on Friday, April 7 at 11:59am. Late assignments will not be accepted. Submit early and often.
- You are permitted to study with friends and discuss the problems; however, *you must write up you own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.
- We require that all homework submissions are prepared in Latex. If you need to draw any diagrams, however, you may draw them with your hand. Please use *a new page to begin each answer.*

PROBLEM 1 *Negative Cycles*

The Bellman-Ford algorithm can be made to detect a negative cycle by running for one more iteration and seeing if any of the vertices have a shorter path. In this problem, we will prove this observation. Let $d(v,k)$ be the length of the shortest path from the source s to v using at most k edges.

(a) Prove that for any edge $(u,v) \in E$ with weight $w(u,v)$, we have $d(v,k+1) \leq d(u,k) + w(u,v)$.

Solution:

(b) Assume that there is a negative cycle $v_1, v_2, \ldots, v_t, v_1$ i.e. $w(v_1,v_2) + w(v_2,v_3) + \ldots + w(v_t,v_1) < 0$. Prove that there exists a vertex v_i on the cycle for which $d(v_i,k+1) < d(v_i,k)$. Hint: add up what you proved in the previous part for all edges in the negative cycle.

Solution:

PROBLEM 2 *Another All-pairs short paths*

Let $G = (V,E)$ be a directed graph with edge weights $w(e)$ and no negative cycles.

(a) Write one sentence that explains the variable $d(i,j,k)$ used in the Floyd-Warshall all-pairs shortest path algorithm discussed in class.

Solution:

(b) State the run time of the All-pairs shortest path algorithm discussed in class.

Solution:

(c) Consider the following algorithm.

procedure NEWALLPAIRS(G,w)
    Add a new node s' to G. Add edges of weight 0 from s' to every vertex $v \in V$.
Call this new graph G'.
    Run BELLMANFORD(G',s') to produce shortest path lengths $\delta(s',v)$.
    For each $e = (x,y) \in E$, set $w'(e) \leftarrow w(e) + \delta(s',x) - \delta(s',y)$
    For each $v \in V$, run DIJKSTRA(G,v,w') to compute $\delta(v,w)$ for all $w \in V$.
    Set $d_{v,w} \leftarrow \delta(v,w) - \delta(s',v) + \delta(s',w)$

8: **end procedure**

We aim to analyze why this algorithm works. The first step is to argue that the new edge weights w' that are defined in step (3) are all non-negative.

Prove that for all $e = (x, y) \in E$, $w'(e) \geq 0$. Hint: consider relation among $\delta(s', y), \delta(s', x), w(e)$. The "tense" edges in Dijkstra's algorithm also used a similar relation.

Solution:

(d) This explains why we can use the fast Dijkstra algorithm with weight w' in step (4) to compute shortest paths from node $v \in V$ to all other nodes in the graph. However, we must argue that the shortest paths under w' and under w will be the same shortest path.

Prove that for any pairs of nodes $u, v \in V$, if p is a shortest path from u to v with respect to weight function w', then p is also a shortest path from u to v with respect to weight function w. Hint: consider an arbitrary path p' and compare its lengths with respect to weight functions w' and w.

Solution:

(e) What is the running time of NewAllPairs in terms of V and E? When does this algorithm run faster than the Floyd-Warshall all-pairs algorithm discussed in class?

Solution: