

Requizzing Period 1- Standard 3

Due Date October / 11th / 2021
Name **Michael Ghattas**
Student ID **109200649**

Contents

1	Instructions	1
2	Honor Code (Make Sure to Virtually Sign)	2
3	Standard Dijkstra	3
3.1	Problem 2	3

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to \LaTeX .
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this \LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students**. **Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.

- My submission is in my own words and reflects my understanding of the material.
- I have not collaborated with any other person.
- I have not posted to external services including, but not limited to Chegg, Discord, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

I agree to the above, Michael Ghattas.

□

3 Standard Dijkstra

3.1 Problem 2

Problem 2. Suppose we are given a finite, connected, and weighted graph $G(V, E, w)$, where the edge weights are non-negative. We define the *weight* of a path P to be the *product* (note: product, *not* sum!) of the edge weights along P . Fix vertices s, t . Our goal is to find a minimum-weight path from s to t .

- (a) Suppose we construct a new graph $H(V, E, w')$ that is identical to G , with the exception that $w'((x, y)) = \log(w((x, y)))$ for all edges (x, y) . That is, $V(H) = V(G)$ and $E(H) = E(G)$. So we have the same underlying graph, with the only difference being the edge weights. You may take as fact that P is a minimum-weight s to t path in G if and only if P is a shortest s to t path in H .

Suppose now that we run Dijkstra's algorithm on H , in order to find a shortest path from s to t in G . Is this approach valid? Justify your reasoning.

Answer:

No. Since we are given a finite, connected, and weighted graph $G(V, E, w)$, where the edge weights are non-negative, and we assume 0 is treated as a *non-negative*, we need to be concerned about any edges on G having a weight equal to 0, because any additional edge added to the path will be multiplied by 0, causing the entire minimum-weight path P from s to t to be equal to 0 on G . Furthermore, we know that the new graph $H(V, E, w')$ is identical to G , with the exception that $w'((x, y)) = \log(w((x, y)))$ for all edges (x, y) . It is important to note that $(\log 0) = -\infty$, and $(\log 1) = 0$. Due to the fact that we define the *weight* of a path P to be the *product* (note: product, *not* sum!) of the edge weights along P . This means if there is a single edge weight set to 0, *i.e.* $w'((x, y)) = \log(0) = -\infty$, or 1 *i.e.* $w'((x, y)) = \log(1) = 0$, any additional edge added to the path could be multiplied by $-\infty$ or 0, and therefore causing the entire minimum-weight path P from s to t to be equal to $-\infty$ or 0 on H . This will create a conflict in the priority queuing of the algorithm, due to edge(s) with non-positive weight values in G or H , and respectively being unable to identify the correct minimum-weight path from s to t on H and as a result G as well. \square

- (b) Suppose now that the edge weights of G are all positive. That is, $w((x, y)) > 0$ for all edges $(x, y) \in E(G)$. Let $H(V, E, w')$ be the graph corresponding to G , as defined in part (a). Is it now a valid approach to run Dijkstra's algorithm on H , in order to find a shortest path from s to t in G ? Justify your reasoning.

Answer:

No. We know that the new graph $H(V, E, w')$ is identical to G , with the exception that $w'((x, y)) = \log(w((x, y)))$ for all edges (x, y) . It is important to note that $(\log 1) = 0$. Due to the fact that we define the *weight* of a path P to be the *product* (note: product, *not* sum!) of the edge weights along P . This means if there is a single edge weight set to 1 *i.e.* $w'((x, y)) = \log(1) = 0$, any additional edge added to the path would be multiplied by 0, and therefore causing the entire minimum-weight path P from s to t to be equal to 0 on H . This will create a conflict in the priority queuing of the algorithm, due to edge(s) with weight values = 0 in H , and respectively being unable to identify the correct minimum-weight path from s to t on H and therefore G . \square

- (c) Give conditions on the edge weights of G , so that it suffices to run Dijkstra's algorithm on H , in order to find a minimum-weight path from s to t in G . Clearly explain why your conditions are correct.

Answer:

Since \log is a continuous and strictly increasing function, set all the edges of G to weight values > 1 i.e. $w((x, y)) > 1$, and therefore all the edge weight values of H will be > 0 i.e. $w'((x, y)) > 0$. This will avoid any additional edge added to the path to be multiplied by 0, and therefore avoid causing the entire minimum-weight path P from s to t to be ≤ 0 on H or $= 0$ on G . Thus, avoiding the issues we faced in parts (a) and (b), where we faced a conflict in the priority queuing of the algorithm, due to edge(s) with non-positive weight values in H , and respectively being unable to identify the correct minimum-weight path from s to t on H and G . \square