Assignment 9

SFWRENG 2CO3: Data Structures and Algorithms–Winter 2024

Deadline: March 25, 2024

Department of Computing and Software
McMaster University

Please read the *Course Outline* for the general policies related to assignments.

Plagiarism is a *serious academic offense* and will be handled accordingly.
All suspicions will be reported to the *Office of Academic Integrity*
(in accordance with the Academic Integrity Policy).

This assignment is an *individual* assignment: do not submit work of others. All parts of your submission *must* be your own work and be based on your own ideas and conclusions. Only *discuss or share* any parts of your submissions with your TA or instructor. You are *responsible for protecting* your work: you are strongly advised to password-protect and lock your electronic devices (e.g., laptop) and to not share your logins with partners or friends!

If you *submit* work, then you are certifying that you are aware of the *Plagiarism and Academic Dishonesty* policy of this course outlined in this section, that you are aware of the Academic Integrity Policy, and that you have completed the submitted work entirely yourself. Furthermore, by submitting work, you agree to automated and manual plagiarism checking of all submitted work.

Late submission policy. Late submissions will receive a late penalty of 20% on the score per day late (with a five hour grace period on the first day, e.g., to deal with technical issues) and submissions five days (or more) past the due date are not accepted. In case of technical issues while submitting, contact the instructor *before* the deadline.

Problem 1. A regional government wants to improve their existing infrastructure between a collection of towns T. In specific, the government want to build a minimum number of roads such that there is a route from each town to each other town. The government has been advised by a dubious consultant that in the resulting road network, the number of users of a given road is independent of the presence of alternative routes.

The regional government wants to minimize the number of roads it has to built to ensure that one can travel from one town to the other. Furthermore, the government wants to maximize the benefits of the road network by maximizing the number of users of the roads built. Hence, the government wants to only build roads that are expected to be used often. To help the construction plans, the government has asked the dubious consultant to estimate, for each pair of cities, the number of road users that would use the road between these two cities (if that road was built).

Now the regional government is looking for a construction plan for a minimum number of roads connecting all towns that see the highest total usage among them.

P1.1. Model the above problem as a graph problem: what are the nodes and edges in your graph, do the edges have weights, and what problem are you trying to answer on your graph?

P1.2. Provide an algorithm ConstructionPlan to find the minimum number of roads to build. Explain why your algorithm is correct.

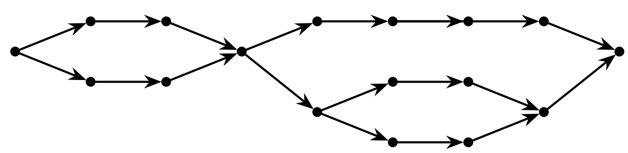
P1.3. Explain which graph representation you used for your algorithm and what the complexity of your algorithm is using this graph representation.

P1.4. What is the worst-case complexity of your solution if you use the other graph representation? Explain your answer.

Problem 2. A directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, s, t)$ with $s \in \mathcal{N}$ the *source* and $t \in \mathcal{N}$ the *target* is a series-parallel graph if it can be constructed inductively using the following rules.

1. An *elementary* series-parallel graph is a single edge from s to t.
2. The *series*-construction. Let $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{E}_1, s, c)$ and $\mathcal{G}_2 = (\mathcal{N}_2, \mathcal{E}_2, c, t)$ be two series-parallel graphs with only the node c in common ($\mathcal{N}_1 \cap \mathcal{N}_2 = \{c\}$). The graph $\mathcal{G} = (\mathcal{N}_1 \cup \mathcal{N}_2, \mathcal{E}_1 \cup \mathcal{E}_2\}$ is a series-parallel graph.
3. The *parallel*-construction. Let $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{E}_1, s_1, t_1)$ and $\mathcal{G}_2 = (\mathcal{N}_2, \mathcal{E}_2, s_2, t_2)$ be two series-parallel graphs without nodes in common ($\mathcal{N}_1 \cap \mathcal{N}_2 = \emptyset$) and let $s, t \notin (\mathcal{N}_1 \cup \mathcal{N}_2)$ be two fresh nodes. The graph $\mathcal{G} = (\mathcal{N}_1 \cup \mathcal{N}_2 \cup \{s, t\}, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \{(s, s_1), (s, s_2), (t_1, t), (t_2, t)\})$ is a series-parallel graph.

The following figure illustrates a series-parallel graph.



Now assume we have a series-parallel graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, s, t)$ with an edge-weight function $\mathit{weight} : \mathcal{E} \to \mathbb{Z}$ (here, $\mathbb{Z}$ are the integers, which includes negative numbers). We note that series-parallel graphs are relatively simple structures.

P2.1. Write an algorithm to compute the single-source shortest paths from the source s to all nodes $n \in \mathcal{N}$ in $\mathcal{O}(|\mathcal{N}| + |\mathcal{E}|)$.

P2.2. Explain why your algorithm is correct.

P2.3. Explain which graph representation you used for your algorithm and why your algorithm has the stated complexity.

P2.4. What is the worst-case complexity of your solution if you use the other graph representation? Explain your answer.

Assignment Details

Write a report in which you solve each of the above problems. Your submission:

1. must start with your name, student number, and MacID;
2. must be a PDF file;
3. must have clearly labeled solutions to each of the stated problems;
4. must be clearly presented;
5. must *not* be hand-written: prepare your report in LaTeX or in a word processor such as Microsoft Word (that can print or export to PDF).

Submissions that do not follow the above requirements will get a grade of zero.

Grading

Each problem counts equally toward the final grade of this assignment.