Assignment 10

SFWRENG 2CO3: Data Structures and Algorithms–Winter 2024

Deadline: March 31, 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. Consider we want to build McMaster Maps, the best online route planning tool in existence. The developers of McMaster Maps have decided to represent road information in terms of a massive graph in which nodes are crossing and the edges are the roads between crossings.

The developers of McMaster Maps have determined that McMaster University is the *only destination that matters*. Hence, McMaster Maps will be optimized toward computing the directions to McMaster University. To do so, McMaster Maps maintains *single-sink shortest path index* that maintains the shortest path from any node to McMaster University. This index is represented by the typical *path* and *cost* arrays as computed by either Dijkstra or Bellman-Ford.

Once in a while, an update to the road network happens: the weight of a single edge changes (this can represent adding and removing edges: addition changes the weight of the edge from ∞ to a numeric value and removing changes the weight of the edge from a numeric value to ∞).

P1.1. Given the road network as a graph G, the *shortest path index*, and the edge that was changed, write an algorithm that determines whether the shortest path index is still valid. You may assume the graph is already updated. Argue why your algrotihm is correct. What is the complexity of your algorithm?

P1.2. Assume the shortest path index is no longer valid: provide a modification of Dijkstra that restores the index to a valid state without recomputing all shortest paths. You may assume the graph is already updated. Argue why your algorithm is correct.

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

Hint: Express the complexity in terms of the number of nodes affected by the change. For example, use a notation in which C is the number of nodes affected by the edge change, $incoming(C)$ the incoming edges of C, and $outgoing(C)$ the outgoing edges of C.

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

Problem 2. Consider a company managing many servers placed all over the world. The company wants to add network connections between a minimal amount of servers to ensure that there is a path of communication between all pairs of servers.

While researching this problem, the company was advised that some connections can be built more reliable than others: according to the consulted contractors, the probability that a connection (m, n) between servers m and n will work at any given time is $p(m, n)$ (we have $p(m, n) = p(n, m)$).

The company wants to *minimize* the number of connects, while *maximizing* the probability that all servers are connected to each other at any given time. We will help the company out in their challenge to figure out which connections they need to built.

P2.1. 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?

P2.2. Provide an algorithm NetworkPlan to find the network connections to build. Explain why your algorithm is correct.

P2.3. Explain which graph representation you used for your algorithm and what the complexity of your algorithm is using this graph representation.

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.