

# Assignment 8

## COMPSCI 2CO3: Data Structures and Algorithms–Fall 2025

Deadline: November 28, 2025

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.* We provide a general *shared grace period* for all assignments of 5 days (120h) that students can utilize whichever way they seem fit without further notice or approval. For example, a student can hand in *one* assignment 100h late, or hand in 4 assignments 20h late (as long as the total stays below 120h). Beyond the above grace period, we require students to *submit on time*. Late submissions are not accepted and will result in a grade of zero. The above grace period *does not stack* with MSAFs or SAS accommodations.

**Problem 1.** In the future, we will use transporters to move from place to place. Unfortunately, linking two transports together is highly costly. Hence, only a limited number of links exist between any pair of transporters and only these linked pairs allow fast travel between them: travel between linked transporters always only takes 5 s.

The planner of TransporterCity often takes the transportation network to move from their home to the office. Currently, the planner has the budget to built one more link between the existing transporters. To avoid bias, the planner wants to make sure that adding the link *does not improve their own travel time* (in the number of transporter steps).

The planner has traveled back in time to ask us to provide a method for suggesting a transporter pair that are not yet linked together, but can be linked together *without* improving the travel situation of the planner.

- 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 efficient algorithm CONSTRUCTIONPLAN that computes a recommendation to the planner in  $O(|\mathcal{N}| + |\mathcal{E}|)$ . You may assume a link can be added that does not improve the situation for the planner itself.
- 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.** Consider an entertainment park in which attractions are connected via a complex network of trails. Seeking for more revenue, the park manager discovered microtransactions (via the spending habits of their children). The manager loved the concept: using microtransactions, the park can extort more money out of their visitors! As such, the manager decided to “lock” all trails in the park and visitors must now “unlock” a path before they can travel that trail using a small transaction. Once unlocked, you can take the trail as often as you want during your stay in the park.

Unfortunately, we are addicted to all attractions and strapped for cash. Hence, we only want to “unlock” a minimum number of trails with a minimum total cost.

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 UNLOCKCHOICE to find the trails to unlock.

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.

Unfortunately, the park manager learned that too many people where minimizing the number of trails “unlocked” using our UNLOCKCHOICE tool. Hence, the manager decided to make trail unlocks one-use items: if you want to traverse a trail  $n$  times, you have to pay  $n$  times.

P2.5. Assume that in the old situation (in which you could visit an unlocked trail multiple times), UNLOCKCHOICE provided a solution that would cost you  $x$  CAD\$. Show how to visit all attractions using at-most  $2x$  CAD\$.

## 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  $\text{\LaTeX}$  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.