Practical Task 8.1

(Pass Task)

Submission deadline: 10:00am Monday, September 16
Discussion deadline: 10:00am Saturday, September 28

General Instructions

In this task, answer all the following questions and complement each answer with a detailed explanation.

1. Given a graph $G = \langle V, E \rangle$, what is to be the running time of the ***depth-first search*** algorithm, as a function of the number of nodes $n = |V|$ and edges $m = |E|$, if the input graph is represented by an ***adjacency matrix*** instead of an *adjacency list*?
2. Kolade is at a train station in a foreign town. He wants to select a hotel that has the *maximum number of shortest paths* from the train station. He thinks that this should reduce the risk of getting lost. Suppose he gives you a city map represented via a graph $G = \langle V, E \rangle$ with $n = |V|$ locations and $m = |E|$ edges connecting the locations. Each edge connecting a pair of directly connected locations has a *unit distance*, say 1. Help Kolade to find a proper hotel by designing a $O(n + m)$ runtime algorithm that finds the number of shortest paths between the train station, located at node s and every hotel on the map. Note that Kolade expects you to convince him that your algorithm is correct and it does find all possible shortest paths. Your solution can be in the form of a pseudocode.
3. A communication network, such as the Internet, can be modelled as an undirected graph $G = \langle V, E \rangle$. Here, the vertices V are the computers on the network, and the edge set E consists of one edge for each pair of computers that are directly connected. We assume that the edges of G are undirected, that is, if there is a direct connection from computer u to computer v, then there is also a direct connection from computer v to computer u.

   It is highly desirable for a communication network graph to be ***connected***, so that every computer on the network can communicate, possibly through a series of relays, with any other computer. But networks can change, with some computers failing and other computers being added to the network. It is useful to have a *testing algorithm* that collects information about the current network graph (vertices and edges) at designated times, and determines properties related to connectivity.

   Describe, in words and pseudocode, a ***testing algorithm*** that given an undirected graph $G = \langle V, E \rangle$ representing the current network decides whether or not the network is connected. A graph (network) is ***connected*** if there is a path from any node to any other node in the graph. You may assume that G is given in *adjacency list* format. Your algorithm must run in $O(n + m)$ time, where $n = |V|$ is the number of computers on the network, and $m = |E|$ is the number of connecting edges.

Further Notes

- You will learn how to complete this task by reading the relevant sections of chapters 14.1-14.3 of the course book “Data Structures and Algorithms in Java” by Michael T. Goodrich, Irvine Roberto Tamassia, and Michael H. Goldwasser (2014). You may access the book on-line for free from the reading list application in CloudDeakin available in Resources → Additional Course Resources → Resources on Algorithms and Data Structures → Course Book: Data structures and algorithms in Java.

Marking Process and Discussion

To get this task completed, you must finish the following steps strictly on time:

- Submit your answers to the task via OnTrack submission system. You may submit a hand-written and then scanned document, but ensure that the text and figures are very clear to read. Note that this is a theoretical task, thus we do not expect you to write any program code.
- Meet with your marking tutor to explain your solutions. When the solutions are hand-written, do not forget to bring them with you. Cloud students must record a short video explaining their work and use a sort of white-board, e.g. a graphical editor, or a scanned document with the answers.
- Answer all additional (theoretical) questions that your tutor may ask you. Questions are likely to cover lecture notes, so attending (or watching) lectures should help you with this compulsory interview part. Please, come prepared so that the class time is used efficiently and fairly for all the students in it. You should start your interview as soon as possible as if your answers are wrong, you may have to pass another interview, still before the deadline. Use available attempts properly.

Note that we will not check your solution after the submission deadline and will not discuss it after the discussion deadline. If you fail one of the deadlines, you fail the task and this reduces the chance to pass the unit. Unless extended for all students, the deadlines are strict to guarantee smooth and on-time work through the unit.

Remember that this is your responsibility to keep track of your progress in the unit that includes checking which tasks have been marked as completed in the OnTrack system by your marking tutor, and which are still to be finalised. When marking you at the end of the unit, we will solely rely on the records of the OnTrack system and feedback provided by your tutor about your overall progress and quality of your solutions.