ADJ: Using Graphs for Contact Tracing

1 Background

Learning Objectives:

- LO1: Apply knowledge of undirected graph algorithms to propose a solution to a graphical problem.
- LO2: Explores multiple solution paths (EM@FSE B)
- LO3: Suspends initial judgment on new ideas (EM@FSE D)
- LO4: Articulates the idea to diverse audiences (EM@FSE M)

In this assignment, we will again solve a problem using the ADJ framework, but now, we have a context to consider. When initially constructing a solution to a problem, we sometimes (in school) omit over the recognition that solutions need to be created and deployed. Both of the aspects add constraints to our solutions that are needed to move our solutions from theory into practice. Your ADJ-style solution should be structured in three pieces: Analysis, Design, and Justification. See previous ADJ material for more information.

As you develop your solution, it is a good idea to consider some of the things that can go wrong. In previous assignments, this might have something like an edge case for a particular type of input. Although those issues might still apply in this type of problem, you now have to consider issues that from needing to enact your solution: computational is cheap, but if a system requires an action to occur, that action may have a cost.

1.1 Basic Expectations

As a base expectation, your submissions must demonstrate both attention to instructions, and professionalism. Specifically (and exhaustively), we require the following to assign a non-zero grade to a submission:

1. The author of the submission must be clearly defined on the front page of the document.
2. Answer subsections must be clearly labeled as analysis, design, or justification.
3. Solutions must be in the spirit of the problem. Do not submit solutions to some "clever" edge case of the problem(s).
4. Proper spelling and grammar. Assignments must not have more than three spelling errors per page, or more than one major grammar error per page (which distracts from readability).

1.2 Problem-Solving Requirements

1. Any assumptions you make in Analysis should be both explicitly stated to be assumptions, and reasonable from the prompt.
   (a) Assumptions may look different than previous problems. For example, instead of making assumptions that an input will have a certain format, you may need to instead make assumptions like: the development of this software will be adequately resourced, or that you will have enough support staff on hand to perform an activity. Note also that it will be harder to make reasonable assumptions - that is fine. Your task in more complex problems is to determine and state those assumptions to the best of your ability (and limited knowledge), and then to indicate potential problems (i.e., risk) there might be with your analysis.

2. **Use paragraphs as appropriate, if you give nothing but bullet points then your solution is likely a summary, not a solution, and will not be worth much credit.**

3. The analysis should not be biased towards any particular solution. It must demonstrate that you are making an effort to suspend your initial judgment on new ideas/solutions (EM@FSE D).

4. Do not submit anything that you do not understand, or which you do not think actually works. Your explanation must convince the reader (whose background may be different than your's) that you know and understand what is happening. (EM@FSE M). Consider what types of audiences would read your solution in practice: who are the stakeholders?

2 Problem (25 points)

[Acuña] Consider the following scenario: *you are developing a system to perform contact tracing to notify individuals who may have been exposed to an infectious disease. You are initially provided with lists of people that have been at the same place at the same time. For example: all the people who visited the same restaurant within a one hour window. If someone is sick, you want to notify everyone that was at the shared location, and then people who didn't visit that location but who were in contact with someone who was.*

Of course, your solution won't exist in a vacuum, you need to take into consideration some factors of the context in which your work will be used:

1. *The priority is stopping the spread of the disease. This means that how long your approach takes to run, or how much space it uses, is less important than producing a usable answer. It also means that you want to err on the safe side when notifying people (i.e., it is better to inform someone who is not sick, rather than not inform someone who is sick).*

2. *The resulting actions of your solution (list of people to contact, based who has been exposed), exists within the real world. For example: we can't automatically notify an infinite number of people at once since no real world systems allow such a thing. Also: although notification might happen quickly, it does take some amount of time. Further:*

   (a) *Recognize that "contact" is an open-ended objective. Your system may have access to emails or phones for people, but what about people who only have a mailing address? Someone will need to be sent to talk to them in person.*

   (b) *A thought: if we simply text someone with a message, is that enough? To properly notification someone, don't we need to make sure they actually read/received/understood the message?*

3. *The people who do the contact tracing in the field are not perfect. They may mistakes. The data the system processes will not be perfect either. You need to propose a solution that has some level of resiliency to these factors and/or discussion of how your system will still be able to accomplish its goal.*

Design an approach to doing contact tracing with graphs. **Analyze** the problem, **design** a high level approach, and **justify** the solution ability to meet your requirements from analysis.

How might you approach this problem? Hint, that you may have guessed: start by considering what type of graph (why undirected?) is appropriate, what nodes represent, and what edges represent. Before going on to thinking about which algorithm (BFS, DFS, or topological sort) is useful, consider how some of the factors above play into your choices. The problem statement provides an initial set of requirements. As your first step, you'll produce a set of concrete requirements (metrics?) during your analysis. Consider using a diagram to show how you structure the graph within the problem.

3 Submission

The submission for this assignment has one part: a write up. The file should be attached to the homework submission link on Canvas.

Writeup: Submit the ADJ answers in PDF format. Please name your file as "LastNameADJ3.pdf" where the last names are given in alphabetic order (e.g. "AcunaADJ3.pdf").