COMP 4601A
Fall 2022 - Lab #9

Objectives

The goal for this lab is to generate recommendations using the graph-based approach covered in the Sparsity and Cold Start Problem lecture (week #11).

Demonstrating/Submitting

There will be several ways to receive credit for completed labs, outlined below:
1. Attend an in-person lab or online office hours and demonstrate your completed lab before the deadline. You will have to show that the goals of the lab have been completed and answer some questions about the lab and your code (see the lab reflection questions for some examples). Your grade will depend on the level of completion, as well as the quality of your design and answers. Only one partner is required for demonstration, though both are encouraged to take part. If you demonstrate your lab this way, you don’t need to submit anything on Brightspace.
2. Record a video demonstration that is ~5-10 minutes long. Ensure that your discussion in the video makes it clear that you have understood the content that the lab covers and that you demonstrate all of the required functionality. Submit a ZIP containing a copy of your code (don't include database files, etc.), your answers to the lab reflection questions, and a copy of your demonstration video (either link to a public URL in your README or include the video file directly) to Brightspace.

If you are working with a partner, only one of you should make a submission. Include a README indicating the names of both partners.

Lab Description

The data provided for this lab follows a similar format to past labs. The main change within the provided data is that each rating value is now either a 1 (liked) or 0 (unknown). The goal for this lab is to generate an output of recommended items for User1 using the graph-based approach discussed in the Sparsity and Cold Start Problem lecture. You are only required to support searching for paths of length 3 for this lab. For each of the provided input files, your implementation should output a sorted list of recommended products. The output data should also indicate the total number of paths that led from the target user to the recommended item.

The included expected-output.txt file contains the sorted list and number of expected paths for each of the provided input files.