CSE416A

HOMEWORK 1

M. Neumann

Due: FRI 20 SEPT 2024 (11:59PM)

SUBMISSION INSTRUCTIONS

- **use of AI to create solutions**
  - The use of AI tools (such as (but not limited to) ChatGPT, Claude, Gemini, or Copilot) to generate solutions (code or written answers) for labs, worksheets, or hw assignments is **not** permitted unless explicitly stated otherwise on the assignment.
  - We will use AI detection tools to identify work that may be in violation of course policies. We may request that students show evidence of their work process (such as notes or earlier versions of final work) should we suspect inappropriate use of AI tools.
- **written work**
  - needs to be submitted electronically in *pdf format* via GRADESCOPE
  - start every problem on a *new page*
  - we prefer *typed submissions*, e.g., using LaTeX (if we cannot read your handwriting we cannot give you credit)
- **code (Jupyter notebook)**
  - needs to be submitted in form of a submission to the corresponding GRADESCOPE programming assignment (instructions can be found on the course webpage)
  - make sure to change the *file name(s)* including your name(s) and follow the *formatting instructions* provided in the notebook (otherwise we cannot grade your submission)
- **group work** (up to 2 students – not for HW0)
  - make a **goup submission** via GRADESCOPE (one submission per team) for both written work and code submission

PIAZZA

We use Piazza for all course and homework related announcements. Ask **all questions** on Piazza using the appropriate tags.

GRADING RESULTS AND REGRADES

Grades will be uploaded to Canvas and detailed grading comments will be provided via GRADESCOPE . You will be notified via GRADESCOPE when the grades are published. All regrade requests need to made via GRADESCOPE **within one week** of this announcement.

Problem 1: Data Structures for Graphs (15%)

Consider the following stack-overflow network, where an edge (a, b) in the network means that person a endorsed an answer from person b on a Java-related question. Note that this network is *directed*: `https://wustl.box.com/s/8hge7d3navi3ediqk6zp1mj6ihoqdefv`
See `hw1_p1_YourName.ipynb` for (additional) instructions.

1.1 Argue whether the stack-overflow graph is dense or sparse.

1.2 How many entries does the *adjacency matrix* for this network have?

1.3 How many entries (think number of integers) does the *edge list* for this network have?

1.4 How many entries does the *adjacency list* for this network have?

1.5 Considering both *data storage* (think memory allocation at runtime) and *data analysis* (think lookup of information within the data structure, as well as iteration) discuss the advantages and disadvantages of each of the data structures. Use your results form the previous parts and code in `hw1_p1_YourName.ipynb` for justifications. HINT: A table makes a good overview representation for this.

Problem 2: Complete graphs and shortest paths (15%)

[Pen-and-paper] A *complete graph* $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ edges is defined as the graph with the maximum number of edges. Assume that G is *undirected* and has no *self loops*.

2.1 What is the number of edges m for a complete graph (as a function of n)?

2.2 What is the average node degree (as a function of n)?

2.3 How could you use a complete graph to model social interactions between people? Think about a suitable type of graph and interpret the meaning of its data.

2.4 Let P denote a shortest path between the nodes u and v (that are at a distance greater than 2 from each other) in a network $\tilde{G}$, and w be another node on this path P. $\tilde{G}$ is not necessarily a complete graph. If Q denotes the segment of P between u and w, including both u and w (see Figure 1 for an illustration), is it true that Q is a shortest path between the nodes u and w? If yes, give a short proof. If no, explain with an example.

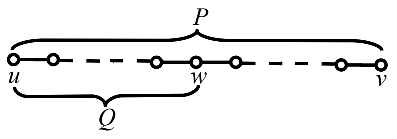


Figure 1: Problem 1.

PROBLEM 3: Average node distance (30%)

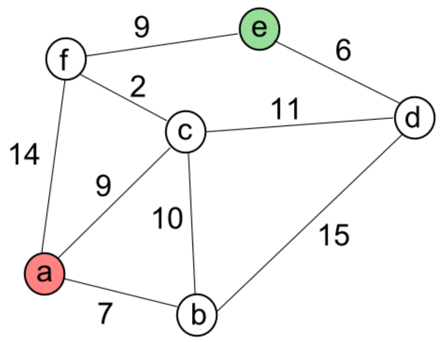
When we think about a single aggregate measure to summarize the distances between the nodes in a given graph, there are two natural quantities that come to mind. One is the *diameter*, which we define to be the maximum distance between any pair of nodes in the graph. Another is the *average distance*, which – as the term suggests – is the average distance over all pairs of nodes in the graph.

3.1 **[Pen-and-paper]** Perform BREATH-FIRST SEACH (BFS) to compute all distances from node CASE to any other node in the Arpanet (cf. [NCM] Figure 2.3). Show your work to receive maximum credit.

3.2 **[Implementation]** Represent the Arpanet using an edge list and run BFS to verify your computations programatically. You may use NetworkX functions. Add the edge list, your program, and the output of your program to your written submission.

3.3 **[Plot]** Now, compute all pairwise distances between all nodes in the Arpanet and plot the *distance distribution* (`matplotlib.pyplot.hist` will be useful). Your program should be a general purpose program – do not hardcode the distances. You will use this program again in a future assignment. Add the plot to your written submission. HINT: Be careful to not *double count* any edges!

3.4 **[Pen-and-paper]** Perform DIJKSTRA's algorithm to compute the distances from node e to any other node in the network shown below. Follow the algorithm and show your work to receive credit.



3.5 **[Conceptual Understanding]** What is the difference between BFS and DIJKSTRA in terms of graph types?

PROBLEM 4: Representing and Analyzing the Human Disease Network (40%)

See `hw1_p4_YourName.ipynb` for instructions.