ITU Computer Engineering Department
BLG 336E Analysis of Algorithms II, Spring 2022
Project #3
Due June 5, 2022 11:59pm

Problem Definition

One of the key problems in graph theory is to measure the maximum flow between a pair of nodes in a weighted graph. However, existing approaches may generally disregard the stochastic nature of a graph. Some of the nodes may become separated from each other as a result of an edge which might have been removed for some reason. Nevertheless, we can still model the stochasticity in a graph, and calculate the maximum flow distribution, accordingly.

In this homework, we will be sorting out the calculation of the *expected* maximum flow, $\mathbb{E}[f_G]$, of a stochastic graph network, where the graph is denoted by the following notation, $G(V, E, P)$. Here, V is the set of vertices, E is the set of edges, and P is the set of probabilities which describes the probability of an edge to be removed, while each of these probabilities are assumed to be independent. The weights of the edges are demonstrated as $W(e)$ which are composed of non-negative integer values.

In order to calculate of the expected maximum flow, we ask you to implement the **Ford-Fulkerson algorithm** which you may want to reconsider looking up the course notes in case of doubt.

Constraints

- The graphs can be directional.
- The cardinality of vertices is denoted by $|V|$, where $2 \leq |V| \leq 100000$
- $0 \leq W(e) \leq 1000, \forall e \in E, W(e) \in \mathbb{Z}^{0+}$
- If it is not possible to reach from a source node to a target node for a subgraph, maximum flow should be 0.

Sample Cases

In input 1, the first line should contain $|V|$, and the lines between 2 to 5 store the adjacency matrix of G. The probability of destruction for each edge are pointed in lines 7 to 10.

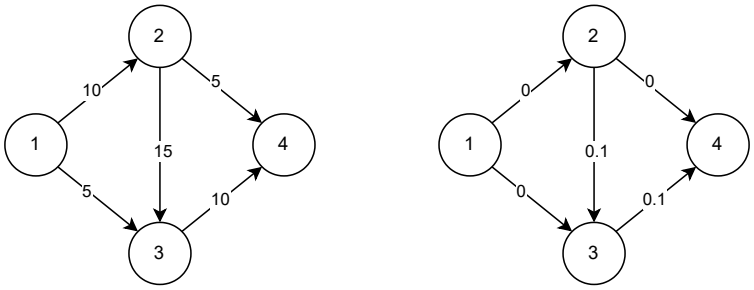

Figure 1: A directed graph where edge weights are shown on left and probabilities of edge removals are shown on right.

Code Listing 1: input1.txt

```
4
0 10 5 0
0 0 15 5
0 0 0 10
0 0 0 0

0 0 0 0
0 0 0.1 0
0 0 0 0.1
0 0 0 0
```

Code Listing 2: Output for Case 1

```
~$ ./homework3 cases/case1.txt 1 4
Probability of occurence: 0.81, Maximum Flow: 15
Probability of occurence: 0.10, Maximum Flow: 5
Probability of occurence: 0.09, Maximum Flow: 10

Expected Maximum Flow from node 1 to node 4: 13.55
```

Based on the attributes of the graph, as shown in Fig. 1, we need to consider four different states, depending on which edge(s) will be disconnected. First, if neither $2-3$ nor $3-4$ becomes disconnected, we calculate the maximum flow of G as 15, and this subgraph may occur with a probability of $(1-0.1)\times(1-0.1) = 0.81$. Secondly, when only the edge between 2 and 3 gets disconnected, the maximum flow is calculated as 10 for this subgraph with a probability of $0.1 \times (1-0.1) = 0.09$. Third, the edge between 3 and 4 may get delinked, and consequently, the maximum flow becomes 5 with a probability of $(1-0.1) \times 0.1 = 0.09$. Finally, if both of these edges get disconnected, the maximum flow for that subgraph of G becomes 5 with a probability of 0.01. Since two of these states are leading to the same maximum flow value, they are merged, and each of the subgraph's probability of occurrences is printed in descending order along with the maximum flow. In addition, the expectation of the maximum flow for graph G is calculated as 13.55 and printed afterwards.

Project Deliveries

- Your C++ Implementation of Ford-Fulkerson working on random graphs [70 pts]
- Report on the project [30 pts]

Report Structure

Your report should include a pseudocode describing your approach to tackle this problem including the algorithmic complexity and a formal definition of expected maximum flow (using mathematical expressions). The report should also cover the following aspects:

- How does network flow are related to the node centrality measures?
- Describe a potential real-life scenario where this sort of stochastic graph modelling and expected maximum flow can be useful.
- Does adding or removing edges *always* alter the maximum flow? If yes, please describe the reason. If not, state your objection with a counterexample.

Submission Rules

- You should write your code in C++ language and try to follow an object-oriented methodology with well-chosen variables, methods, and class names and comments where necessary. You can use the C++ Standard Template Library (STL). You can define multiple classes in a single cpp file or use multiple cpp files with header files. If it is not possible to compile your homework using "**g++ -Werror -Wall main.cpp -o homework3**", make sure to include a MakeFile which makes your code to be compiled with "**make all**" command.
- Also, make sure that your code can be run in the form of **./homework3** [**input_file**] [**src_node**] [**dst_node**].
- If the code is not self-explanatory and does not include adequate comments, a point penalty of upto 10 points will be applied.
- Do not share any code or text that can be submitted as a part of an assignment (discussing ideas is okay).
- Only electronic submissions through Ninova will be accepted no later than deadline.
- You may discuss the problems at an abstract level with your classmates, but you should not **share or copy code** from your classmates or from the Internet. You should submit your **own, individual** homework.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- If you have any questions about the recitation, you can send an e-mail to Caner Özer (ozerc@itu.edu.tr).
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**