CSE208: Data Structures and Algorithms II Sessional

January 2020

Assignment on Maximum Flow and Maximum Bipartite Matching

In this assignment, you need to implement the Ford-Fulkerson method for the maximum flow problem and an algorithm for the maximum bipartite matching based on the Ford-Fulkerson method. Your implementation of the Ford-Fulkerson method should run in $O(E|f^*|)$, where f^* is the maximum flow (or optionally $O(VE^2)$ if you choose augmenting paths using the Edmonds-Karp algorithm) and the bipartite matching algorithm should run in $O(VE)$.

You will use the **Graph** class with an **adjacency list** representation that you implemented previously.

You will find detailed descriptions of these two algorithms in your textbook "Introduction to Algorithms is a book on computer programming" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Chapter 26.

Input / Output Format

You will take input from an input file and print output to an output file. Keep provision of printing output to the console as well (but printing both to file and to console simultaneously will not be necessary).

Input Format (Maximum flow):

The input graph is **directed.**

The first line of input will have two space-separated non-negative integers N and M, the number of nodes and edges in the graph. In the next M lines, there will be three space-separated integers, u, v, c denoting a directed edge (u, v) and its capacity c=c(u,v). ($0 \leq u, v < N$)

The last line will contain two space-separated non-negative integers s, t denoting a source and a sink respectively.

Output Format (Maximum flow):

Print the maximum flow from *s* to *t* found by your algorithm in the given graph in the first line. In the following M lines, you need to print the flow through each edge where each line will contain four integers in the format u v f/c denoting the flow *f* through the edge (u, v) with capacity c.

Input Format (Bipartite matching):

The input graph is **undirected.**

The first line of input will have two space-separated non-negative integers N and M, the number of nodes and edges in the graph. In the next M lines, there will be two space-separated integers, u, v denoting an undirected edge (u, v).

Output Format (Bipartite matching):

Print the number of edges in the maximum bipartite matching found by your algorithm in the given graph in the first line. In the following lines, you need to print the edges picked by your algorithm that correspond to the maximum matching. Each line will contain two space-separated integers, u, v denoting the edge (u, v).

If the graph is not Bipartite, print **"The graph is not bipartite".**

See the sample I/O for further clarification

Sample I/O (Maximum flow)

Input	Output
6 10 0 1 16 0 2 13 1 2 10 1 3 12 2 1 4 2 4 14 3 2 9 3 5 20 4 3 7 4 5 4 0 5	23 0 1 11/16 0 2 12/13 1 2 0/10 1 3 12/12 2 1 1/4 2 4 11/14 3 2 0/9 3 5 19/20 4 3 7/7 4 5 4/4

*From “Introduction to Algorithms is a book on computer programming” by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Chapter 26

Sample I/O (Bipartite matching)

Input	Output
9 8 0 5 1 5 1 7 2 6 2 7 2 8 3 7 4 7	3 1 5 2 6 4 7

*From “Introduction to Algorithms is a book on computer programming” by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Chapter 26

Special Instructions

Write ***readable, re-usable, well-structured, quality*** code. Do not write unnecessary functions or declare unnecessary variables. Delete all unnecessary variables and functions. Give meaningful names to your variables and maintain proper indentations.

You also cannot use any other built-in functions or libraries other than the vector and priority_queue classes.

Please DO NOT COPY solutions from anywhere (your friends, seniors, internet etc.)

Implement the algorithms with your own style of coding. **Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks in the assignment. It is your duty to protect your code.**

Also, be informed that for repeated offense of plagiarism, the departmental policies suggest stricter measures.

Submission Guidelines

1. Create a directory with your 7 digit student id as name.

2. Put the source file only into the directory created in 1.

3. Zip the directory.

4. Upload the zip into moodle.

For example, if your student id is 1705123, create a directory named 1705123. Put your source files(.cpp, .java etc) only into 1705123.zip and upload the 1705123.zip into moodle.

Failure to follow the above mentioned submission guideline will result in some penalty.

Marks Distribution

Implementation of the Ford Fulkerson method	6
Implementation of the bipartite matching algorithm	3
Code readability, cleanliness and maintenance of assignment instructions	1
Total	**10**

Deadline

Deadline is set on 9 October, 2020 (Friday) at 11:55 pm.

THIS IS A HARD DEADLINE. THERE WILL BE NO EXTENSIONS FOR ANY REASON WHATSOEVER. START YOUR ASSIGNMENT EARLY.

For any query, please email atif.bd@gmail.com