Computer Science C73 November 10, 2021
Scarborough Campus University of Toronto

Homework Assignment #7
(worth 6% of the course grade)
Due: November 24, 2020, by 11:59 pm

• ***You must submit your assignment through the Crowdmark system.*** *You will receive by email an invitation through which you can submit your work in the form of separate PDF docucments with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark. Crowdmark does not enforce a limit on the size of groups.* ***The course policy that limits the size of each group to at most two remains in effect:*** *submissions by groups of more than two persons will not be graded.*
• *It is your responsibility to ensure that the PDF files you submit are legible. To this end, I encourage you to learn and use the LaTex typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You are not required to produce the PDF files you submit using LaTex; you may produce it any way you wish, as long as the resulting document is legible.*
• *By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.*[a]
• *For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the assigned sections of the official course textbooks, by referring to it.*
• *Unless we explicitly state otherwise, you may describe algorithms in high-level pseudocode or point-form English, whichever leads to a clearer and simpler description.*
• *Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation.*

[a] "In each homework assignment you may collaborate with at most one other student who is currently taking CSCC73. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. Collaboration involving more than two students is not allowed. **For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), your textbook, and your class notes. You may not consult any other source.**"

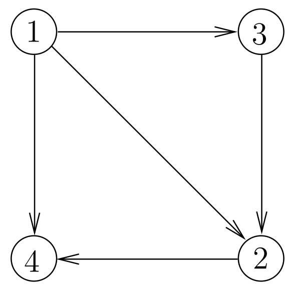
Question 1. (10 marks) A ***bottleneck*** of a flow network $\mathcal{F}$ is an edge of $\mathcal{F}$ such that increasing the capacity of that edge (and of no other edge) increases the value of the maximum flow. Describe an algorithm that, given a flow network $\mathcal{F}$ ***and a maximum flow*** f ***of*** $\mathcal{F}$, returns the set of ***all*** bottleneck edges of $\mathcal{F}$. Your algorithm should run in $O(n + m)$ time, where n is the number of nodes and m is the number of edges of $\mathcal{F}$. Prove that your algorithm is correct and analyze its running time.

Question 2. (10 marks) Let $G = (V, E)$ be a directed graph. For each node u, the ***in-degree*** (respectively, ***out-degree***) of u in G is the number of edges in E that go into (respectively, out of) u. For the remainder of this question we consider only directed graphs with no self-loops, i.e., no edge from a node to itself.

Suppose that we are given a set of nodes V (but no edges), and for each $u \in V$, a pair of non-negative integers, $in(u)$ and $out(u)$; we call this the ***degree-pair*** of u. Is there a directed graph $G = (V, E)$ such that for each $u \in V$, the in-degree of u in G is $in(u)$ and the out-degree of u in G is $out(u)$? If so, we say that the given degree-pairs are ***realizable***, and that the graph G ***realizes*** them.

For example, the degree pairs given below for $V = \{1, 2, 3, 4\}$ is realizable, as confirmed by the graph to the right.

u	$in(u)$	$out(u)$
1	0	3
2	2	1
3	1	1
4	2	0



On the other hand, the degree pairs given below for $V = \{1, 2, 3\}$ is not realizable: each of nodes 2 and 3 has out-degree 2, so each of them must have an edge to node 1, whose in-degree, however, is only 1.

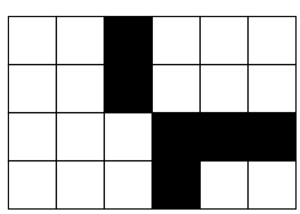
u	$in(u)$	$out(u)$
1	1	1
2	2	2
3	2	2

Give a polynomial-time algorithm that, given a set of nodes V and a pair of non-negative integers $\big(in(u), out(u)\big)$ for each $u \in V$, determines whether these degree-pairs are realizable. If they are, the algorithm should also output a directed graph that realizes them. Prove that your algorithm is correct and analyze its running time.

Hint. Reduce this problem to a max-flow problem: Define a flow network $\mathcal{F} = (G, s, t, c)$ that, in addition to source and target nodes, has two nodes, v_{out} and v_{in}, for each node $v \in V$. It is up to you to determine the edges of G and their capacities. G should be defined in such a manner that the given degree pairs are realizable if and only if the maximum flow of $\mathcal{F}$ satisfies a certain (easy-to-check) property.

Question 3. (10 marks) Consider an $m \times n$ grid (m rows and n columns). Some squares of the grid contain ***tokens*** and others are empty. Each token is identified by the coordinates (i, j) of the square that contains it, where $1 \leq i \leq m$ and $1 \leq j \leq n$. A subset of the set of tokens is called ***irreducible*** if it contains no two tokens on the same row or column. A set consisting of rows or columns is called ***full*** (with respect to the set of tokens on the grid) if each token is on some row or some column in the set.

For example, in the 4×6 grid below, squares containing tokens are shown in black. The following sets of tokens are irreducible: $\varnothing$, $\{(2, 3), (4, 4)\}$, and $\{(1, 3), (3, 6), (4, 4)\}$. The last of these is an irreducible set of maximum cardinality: no irreducible set contains four tokens. The following sets of rows or columns are full, where Ri denotes row i and Cj denotes column j: $\{\text{R1}, \text{C2}, \text{C3}, \text{R3}, \text{C4}\}$, $\{\text{R1}, \text{C3}, \text{R3}, \text{C4}\}$, $\{\text{C3}, \text{R3}, \text{C4}\}$. The last of these is a full set of minimum cardinality: no cover contains only two rows or columns.



Using results discussed in this course prove that the maximum cardinality of an irreducible set of tokens is equal to the minimum cardinality of a full set of rows or columns.

Question to think about <u>but not to include in your solution</u>: Using the insights from the above result and algorithms we have seen in this course, design polynomial-time algorithms that, given the dimensions of the grid and the positions of the tokens, find (a) a cover of minimum cardinality, and (b) an irreducible set of maximum cardinality.