Computer Science CSC373H
St. George Campus

March 10, 2021
University of Toronto

Homework Assignment #5

Due: March 17, 2021, by 11:00 am

- **You must submit your assignment through the Crowdmark system.** You will receive by email an invitation through which you can submit your work. If you haven't used Crowdmark before, give yourself plenty of time to figure it out!
- You must submit a **separate** PDF document with for **each** question of the assignment.
- To work with one or two partners, you and your partner(s) must form a **group** on Crowdmark (one submission only per group). We allow groups of up to three students, submissions by groups of more than three students will not be graded.
- The PDF file that you submit for each question must be typeset (**not** handwritten) and clearly legible. To this end, we encourage you to learn and use the LaTeX typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You can use other typesetting systems if you prefer, but handwritten documents are not accepted.
- If this assignment is submitted by a group of two or three students, for each assignment question the PDF file that you submit should contain:
  1. The name(s) of the student(s) who *wrote* the solution to this question, and
  2. The name(s) of the student(s) who *read* this solution to verify its clarity and correctness.
- By virtue of submitting this assignment you (and your partners, if you have any) acknowledge that you are aware of the homework collaboration policy for this course, as stated in: `http://www.cs.toronto.edu/~sam/teaching/373/#HomeworkCollaboration`.
- 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 (in lectures or tutorials) by referring to it.
- 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.
- The total length of your pdf submission should be no more than 3.5 pages long in a 11pt font.

Question 1. (10 marks) Describe an algorithm that, given any undirected graph $G = (V, E)$, determines the ***minimum*** number of edges that must be removed to disconnect the graph. Your algorithm should work by running the maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges. Prove the correctness of your algorithm.

HINT: The basic idea is to relate the minimum number of edges that must be removed to disconnect the undirected graph G with the capacity of some cuts in some network flows derived from G.

Question 2. (10 marks) Let $S = \{s_1, \ldots, s_n\}$ be the set of Master's students in the Department of Computer Science (DCS), and $C = \{c_1, \ldots, c_m\}$ be the set of graduate courses offered by DCS. C is partitioned into two sets T_1 and T_2, the set of courses offered in the fall and winter term, respectively.

Each course c_i has an enrolment limit of ℓ_i. Each student s_j chooses a set $C_j \subseteq C$ of courses that s_j is interested in taking. (C_j may be any subset of C, including the empty set!)

DCS must assign each Master's student to a set of courses that the student is interested in taking. A student may be assigned to at most 5 courses, at most 3 of which may be in the same term. Furthermore, no course should be assigned more students than its enrolment limit stipulates.

a. DCS receives from the university \$1,000 for each student in each course. How should it assign students to courses so as to maximise its revenue? Express this problem as an instance of the maximum flow problem. Justify your answer.

b. A student can graduate if he/she takes 5 courses. How should DCS assign students to courses so as to maximise the number of students who can graduate? Express this problem as a 0-1 ILP (integer linear program). Justify your answer.

HINT: Among the variables you define, include a variable y_i for each student i, whose intended semantics is that $y_i = 1$ if student i is assigned to five courses, and $y_i = 0$ otherwise. Then introduce constraints (relating this variable to other variables) so that the value of y_i satisfies these semantics.