University of Toronto at Scarborough
CSCC73 - Algorithm Design and Analysis

Assignment #6: Network Flow

Due: March 13, 2023 at 11.59pm This exercise is worth 5% of your final grade.

Warning: Your electronic submission on Gradescope affirms that this exercise is your own work and no one else's, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters, the Code of Student Conduct, and the guidelines for avoiding plagiarism in CSCC73. Late assignments will not be accepted. If you are working with a partner your partners' name must be listed on your assignment and you must sign up as a "group" on MarkUs. Recall you must not consult **any outside sources except your partner, textbook, TAs and instructor**.

1. (10 marks) Suppose you have a flow network. We define a *critical* edge on the network to one that if we decrease the capacity of this edge the maximum flow is decreased. Give an efficient algorithm that finds a critical edge in a network. Be sure to explain the complexity of your algorithm and justify the correctness of your algorithm.

2. (10 marks) A vertex cover of an undirected graph $G = (V, E)$ is a subset of the vertices which touches every edge, ie., that is, a subset $S \subset V$ such that for each edge $\{u, v\} \in E$, one of both of u, v are in S.

   Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. HINT: Consider relating this problem to minimum cuts.

3. (10 marks) Suppose you're looking at a flow network G with source s and sink t, and you want to express the intuition that some nodes are clearly on the "*source side*" of the main bottlenecks; some nodes are clearly on the "*sink side*" of the main bottlenecks; and some nodes are in the middle. However, G can have many minimum cuts, so we have to be careful in how we make this idea precise.

   - We say a node v is *upstream* if, for all minimum s, t-cuts (A, B), we have $v \in A$–that is, v lies on the source side of every minimum cut.
   - We say a node v is *downstream* if, for all minimum s, t-cuts (A, B), we have $v \in B$–that is, v lies on the sink side of every minimum cut.
   - We say a node is *central* if it is neither *upstream* nor *downstream*; *i.e.*, there is at least one minimum s, t-cut (A, B) for which $v \in A$, and at least one minimum cut (A^*, B^*) for which $v \in B^*$.

   Give an algorithm that takes a flow network G and classifies each of its nodes as either upstream, downstream, or central. The running time of your algorithm should be within a constant factor of the time required to compute a *single* maximum flow. You should carefully explain why your algorithm works.