Worth: 7.5% (*best four of the five assignments*) **Due:** *before* 6:00pm on **Fri. 8 March**

Required filename for MarkUs submission: `a3.pdf`
*(Use a **single** file to submit your answers for both Part I and Part II.)*

Remember to write the *full name* and *MarkUs username* of *every* group member (up to three) prominently on your submission.

---

*Please read and understand the policy on Collaboration given on the Course Information Sheet. Then, to protect yourself, list on the front of your submission **every** source of information you used to complete this homework (other than your own lecture and tutorial notes). For example, indicate clearly the **name** of every student from another group with whom you had discussions, the **title and sections** of every textbook you consulted (including the course textbook), the **source** of every web document you used (including documents from the course webpage), etc.*

*For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks **will** be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions.*

---

3. Reducing Edge Capacities

(a) Prove or Disprove: if $N = (V, E)$ is a network, f^* is a maximum flow in N, $e_0 \in E$ is an edge with $f^*(e_0) = c(e_0)$, and N' is the same network as N except that $c'(e_0) = c(e_0) - 1$, then the maximum flow f' in N' satisfies $|f'| < |f^*|$.

(b) Write an algorithm that takes a network $N = (V, E)$, maximum flow f^* in N, and an edge $e_0 \in E$ with $f^*(e_0) = c(e_0)$, and that outputs a maximum flow for network N', where N' is the same as N except that $c'(e_0) = c(e_0) - 1$.

Provide a brief argument that your algorithm is correct and analyze its time complexity. For full marks, your algorithm should be as efficient as possible.

(c) Write an algorithm that takes a network $N = (V, E)$ and that outputs a list of **all** edges $e_1, \ldots, e_k$ with the property that if the capacity of any edge in that list is reduced **by one unit**, the value of the maximum flow in N is also reduced.

Prove that your algorithm is correct and analyze its time complexity.