Due 23:59 Nov 6 (Sunday). There are 100 points in this assignment.
Submit your answers (**must be typed**) in pdf file to CourSys
`https://coursys.sfu.ca/2022fa-cmpt-705-x1/`. Submissions received after 23:59 of Nov 6 will get penalty of reducing points: 20 and 50 points deductions for submissions received at $[00:00, 00:10]$ and $(00:10, 00:30]$ of Nov 6, respectively; no points will be given to submissions after $00:30$ of Nov 6.

1. (Chapter 7 Problem 11 of the text book) 20 points

   The Forward-Edge-Only algorithm computes a flow in a flow networks as follows: it searches for $s-t$ paths in a graph $\tilde{G}_f$ consisting only of arcs e for which $f(e) < c_e$, and it terminates when there is no augmenting path consisting entirely of such arcs (the algorithm may choose a forward arc path arbitrarily, provided it terminates only when there are no forward arc paths). A claim for the Forward-Edge-Only algorithm is that there is a constant $k > 1$ (independent of the particular input flow network), so that on every instance of the Maximum Flow problem, the Forward-Edge-Only algorithm is guaranteed to find a flow of value at least $1/k$ times the maximum flow value (regardless of how it chooses its forward edge paths). Decide whether you think this claim is true or false, and give a proof of either the claim or its negation.

   Answer

   ---

2. (Chapter 7 Problem 17 of the text book) 20 points

   There is a communication network of n nodes carrying data from a source node s to a destination node t. There are k link-disjoint paths from s to t in the network when all links functions formally. Now an attacker disabled a minimum number of links that t is not reachable from s. You are sitting at node s and using `ping` command to find the disabled links (command `ping`(v) tells whether node v is reachable or not from s). Assume that you have the topology information of the network and you can decide which node to `ping` based on the previous results of the `ping` commands. Give an algorithm which uses $O(k \log n)$ `ping` commands to find the disabled links.

   Answer

   ---

3. (Chapter 7 Problem 21 of the text book) 20 points

   A test instance for a WiFi network consists of n laptops $\{c_1, .., c_n\}$ and n access points $\{p_1, .., p_n\}$. A test set T for the test instance is a set of pairs (c_i, p_j) with the following properties:

   (i) if (c_i, p_j) in T then laptop c_i is within the access range of access point p_j,
   (ii) each laptop appears in at least one pair of T, and
   (iii) each access point appears in at least one pair of T.
   (a) Give an example of a test instance for which there is no test set of size n.
   (b) Give a polynomial time algorithm which, given a test instance and integer k, decides whether there is a test set of size at most k.

Answer

---

4. (Chapter 7 Problem 31 of the text book) 20 points

   There are n boxes $1, .., n$, each box i has size $s(i)$. A box i can be put inside box j if $s(i) < s(j)$. For any two boxes i and i' with $s(i) < s(j)$ and $s(i') < s(j)$, both i and i' can not be put inside j if i is not put inside i' or i' is not put inside i. But for a sequence of boxes $i_1, i_2, .., i_k$ with $s(i_1) < s(i_2) < \cdots < s(i_k)$, i_1 can be put inside i_2, then i_2 put inside i_3,.., and finally i_{k-1} put inside i_k. In this case, all boxes $i_1, .., i_{k-1}$ are inside box i_k and only i_k is visible. The nesting arrangement for a set of n boxes to put one box inside another such that the number of visible boxes is minimized. Give a polynomial time algorithm which, given a set of n boxes $1, .., n$ and $s(1), .., s(n)$, solves the nesting arrangement problem.

   Answer

   ---

5. 20 points

Figure 1 gives a flow network G and a function $f : E(G) \to R^+$. The capacity of each arc appears as a label next to the arc, the value assignd to each arc by f is in the box next to the arc.

(a) Is f a flow on G? If yes, why? and give the residual graph G_f w.r.t. f. If no, why?
(b) Implement Ford-Fulkerson Algorithm for the maximum flow problem and show the result of your implementation for G.

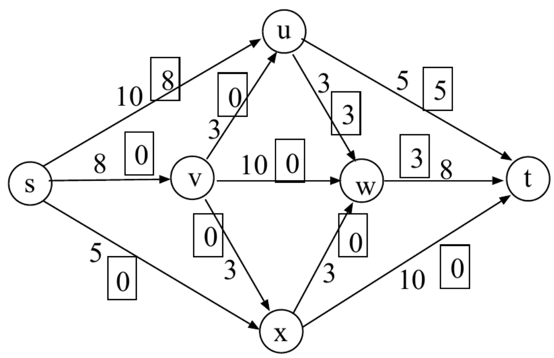

Figure 1: Figure for question 5.

Answer
