CMPSC 465 Data Structures & Algorithms
Fall 2025 Ke Chen and Yana Safonova

HW 7

Due October 27th, 10:00 pm

Instructions: You are encouraged to solve the problem sets on your own, or in groups of three to five people, but you must write your solutions strictly by yourself. You must explicitly acknowledge in your write-up all your collaborators, as well as any books, papers, web pages, etc. you got ideas from.

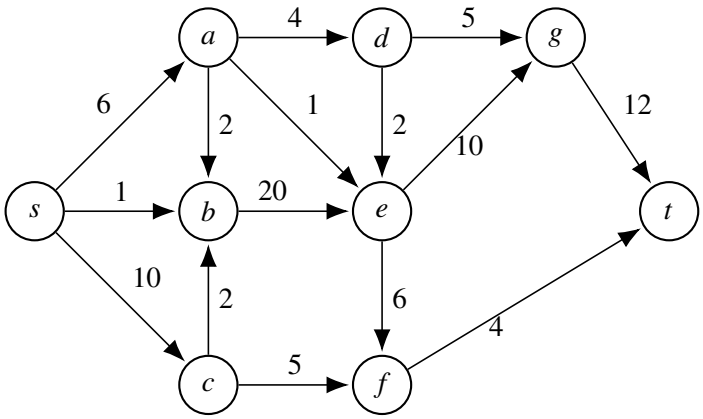
Formatting: Each part of each problem should begin on a new page. Each page should be clearly labeled with the problem number and the problem part. The pages of your homework submissions must be in order. **When submitting in Gradescope, make sure that you assign pages to problems from the rubric. You risk receiving no credit for it if you do not adhere to these guidelines.**

Late homework will not be accepted. Please, do not ask for extensions since we will provide solutions shortly after the due date. Remember that we will drop your lowest three scores.

This homework is due Monday, October 27, at 10:00 pm electronically. You need to submit it via Gradescope. Please ask on Campuswire about any details concerning Gradescope.

1. (25 pts.) **Ford-Fulkerson Algorithm.** Use the Edmonds-Karp algorithm (namely, Ford-Fulkerson where each augmenting path is found by BFS) to find the maximum flow and the corresponding minimum cut in the given $s-t$ flow network. Process neighbors in alphabetical order during BFS. Show the augmenting path and draw the residual graph G_f for each step. Show the final maximum-flow and the minimum cut.

G:



2. (25 pts.) **Max Flow Formulation.** Consider a variation of the standard max-flow problem in a flow network. In addition to the existing edge capacities, each **vertex** has a capacity indicating the maximum amount of flow that is allowed to pass through. Show that this problem can be reduced to the standard max-flow problem and explain any pre-processing or additional data structures used. [Hint: transform this graph with both edge and vertex capacities into a regular flow network with only edge capacities by doubling the number of vertices.]

3. (25 pts.) **Critical Edge.** An edge of an $s-t$ flow network is called *critical* if decreasing the capacity of

this edge results in a decrease in the $s-t$ maximum flow. Give an efficient algorithm that finds a critical edge in a network. Explain the correctness of your algorithm and analyze the running time.

4. (25 pts.) **Application of Network Flow.** Some programs in our university have a complicated set of graduation rules. Suppose a set $C = \{c_1, c_2, \ldots, c_n\}$ of n courses is offered. Each rule is specified by a subset $S \subseteq C$ of courses and an integer x, indicating that a student must take at least x courses from S in order to satisfy that rule. The subsets for different rules may overlap, but each course can be used to satisfy at most one rule. To graduate, a student must satisfy **all** of the rules.

Given a collection of m rules $\{(S_1, x_1), (S_2, x_2), \ldots, (S_m, x_m)\}$ and a set of ℓ courses a student has taken $\{c_{i_1}, c_{i_2}, \ldots, c_{i_\ell}\}$, describe an efficient algorithm to determine whether the student can graduate. Justify the correctness of your algorithm and analyze its running time.

For example, suppose there are 5 courses, $C = \{c_1, c_2, c_3, c_4, c_5\}$, and two rules:

- One must take at least $x_1 = 2$ courses from $S_1 = \{c_1, c_2, c_3\}$.
- One must take at least $x_2 = 2$ courses from $S_2 = \{c_3, c_4, c_5\}$.

Then a student who took $\{c_1, c_3, c_5\}$ cannot graduate, while a student who took $\{c_1, c_2, c_3, c_4\}$ can.