ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2019

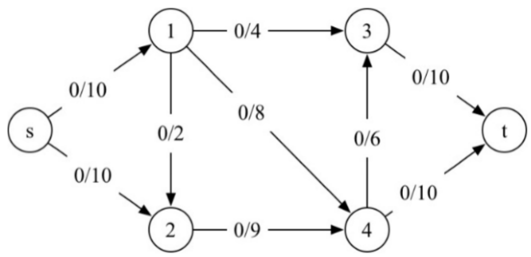
Problem Set 3

Due: 23:59, Apr. 1, 2019

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" of Gradescope, set your FULL NAME to your Chinese name and enter your STUDENT ID correctly.
3. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
4. When submitting your homework, match each of your solution to the corresponding problem number.

Problem 1:

Run the Ford-Fulkerson algorithm on the flow network in the figure below, and show the residual network after each flow augmentation. For each iteration, pick the augmenting path that is lexicographically smallest. (e.g., if you have two augmenting path $1 \to 3 \to t$ and $1 \to 4 \to t$, then you should choose $1 \to 3 \to t$, which is lexicographically smaller than $1 \to 4 \to t$)



Problem 2:

There are n staff members in a company. Each staff member belongs to a department. There are k departments. Now we have to choose one staff member from each department to a company-level committee. In this committee, there must be m_1 number of A-class staff members, m_2 number of B-class staff members, m_3 number of C-class staff members, and m_4 number of D-class staff members (so we have $m_1 + m_2 + m_3 + m_4 = k$). We are given the list of staff members, their home departments, and their classes (A, B, C, D). Describe an efficient algorithm to determine who should be on the committee such that the constraints are satisfied.

Problem 3:

Let $G = (V, E)$ be a flow network with source s, sink t, and integer capacities. Suppose that we are given a maximum flow in G.

(a) Suppose that we increase the capacity of a single edge $(v, u) \in E$ by 1. Give an $O(V + E)$ time algorithm to update the maximum flow.

(b) Suppose that we decrease the capacity of a single edge $(v, u) \in E$ by 1. Give an $O(V + E)$ time algorithm to update the maximum flow.

Problem 4:

Given an $m \times n$ matrix in which every element is a positive integer, you need to select a subset of the elements in the matrix so that these selected elements are not adjacent. We define that element (i, j) is adjacent to elements $(i, j \pm 1)$ and $(i \pm 1, j)$ but is not adjacent to elements $(i \pm 1, j \pm 1)$. Design an efficient algorithm that maximizes the sum of the selected elements.

Problem 5:

A farmer makes F types of foods and D types of drinks for N cows. Each cow has a different preference list of certain types of foods and drinks. Each type of food or drink can only be served to one cow. The farmer gives each cow one type of food and one type of drink and he wants to maximize the number of cows that receive both food and drink types in their preference lists. Design an algorithm to help the farmer.

Problem 6:

Given a network $G = (V, E)$, design a polynomial time algorithm to determine whether G has a unique minimum s-t cut.