ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2020

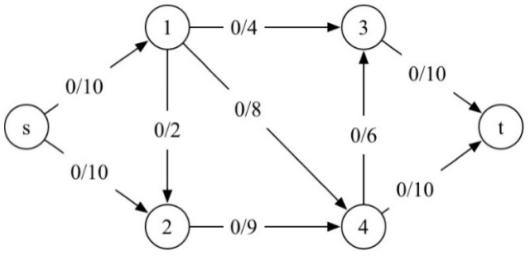
Problem Set 2

Due: 23:59, Apr. 6, 2020

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" of Gradescope, set your FULL NAME to your Chinese name.
3. If you want to submit a handwritten version, scan it clearly.
4. When submitting your homework in Gradescope, 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:

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 4:

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

Problem 5:

We are given an $n \times m$ rectangular table. Denote the cell in the r-th row and the c-th column as (r, c), where $1 \leq r \leq n$ and $1 \leq c \leq m$. A robot is at cell $(1, 1)$ and wants to reach cell (n, m). From cell (x, y), the robot can only move to $(x, y+1)$ or $(x+1, y)$. What's more, some cells contain impassable obstacles and the robot cannot move to these cells. Your goal is to put additional obstacles in the minimal number of cells (except cell (n, m)) such that the robot cannot reach (n, m).

One simple solution is to run DFS twice. However, here you are asked to find a solution using max flow.

Problem 6:

Given a sequence of positive integers $x_1, ..., x_n$, find

1. The length L of the longest strictly ascending subsequence in $O(n^2)$. (Hint: It can be solved by dynamic programming.)

2. How many strictly ascending subsequences of length L can be extracted from the input sequence? These subsequences must be non-overlapping. In other words, if x_i is contained in one subsequence, it cannot be contained in another subsequence.

Note: A subsequence is different from a substring in that a subsequence can be inconsecutive. For example, 1,3,5 is a subsequence of 1,2,3,4,5.