ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2021

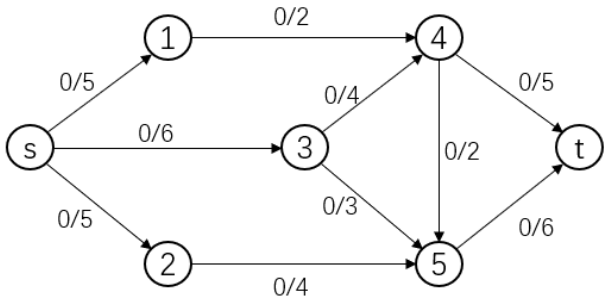
Problem Set 2

Due: 23:59, Mar. 25, 2021

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. You need to write down the augmenting path you pick and show the residual network after each flow augmentation. For each iteration, pick the augmenting path that is lexicographically smallest (for example, if you have two augmenting path $s \to 1 \to 3 \to t$ and $s \to 1 \to 4 \to t$, then you should choose the former because 1-3 is lexicographically smaller than 1-4; if you have two augmenting path $s \to 1 \to 3 \to t$ and $s \to 1 \to 3 \to 4 \to t$, then you should choose the former because 1-3 is lexicographically smaller than 1-3-4).



Problem 2:

There are n players playing games. Every player plays exactly one game with every other player. Ties are not allowed (i.e., each game will be continued until there is a winner). Suppose each player wins v_i times. Give an efficient algorithm to show whether a vector $(v_1, v_2..., v_n)$ is a possible outcome of the game.

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:

Suppose $G = (V, E)$ is a flow network with integer capacities and you already know a maximum flow in G for which every edge has an integer flow. For some edge $e \in E$, the original capacity is c_e.

(a) If c_e is increased to $c_e + 1$. Design an algorithm in $O(|E| + |V|)$ time to calculate the maximum flow after the change.

(b) If c_e is decreased to $c_e - 1$. Design an algorithm in $O(|E| + |V|)$ time to calculate the maximum flow after the change.

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.