ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2023

Problem Set 3

Due: 23:59, April 6, 2023

1. Submit your solutions to the course Blackboard.
2. If you want to submit a handwritten version, scan it clearly.
3. Late homeworks submitted within 24 hours of the due date will be marked down 25%. Homeworks submitted more than 24 hours after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
4. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

Problem 1:

Given an array A, find the longest continuous increasing subsequence in the array and return its length. For example, for the array [1, 3, 2, 4, 7, 6, 8, 9], the longest continuous increasing subsequence is [6, 8, 9] and its length is 3. Design an algorithm, prove its correctness, and analyze its time and memory complexity.

Problem 2:

The school is organizing a spring outing for students to Disney Amusement Park. There are M students, and park has N areas ($M, N \geq 1$). We want to know how many ways there are to divide the students so that each student goes to exactly one area (we allow some areas to not be visited by any students). Note that we only care how many students go to the different areas, not which areas they go to.

For example, if there are 3 students and 2 areas, the output should be 2, because the only divisions are (3,0) and (2,1). Note that here, we consider assigning 2 students to the first area and 1 student to the second, to be the same as assigning 1 student to the first area and 2 to the second; both assignments yield the division (2,1).

Design an algorithm for this problem and analyze its time and memory complexity.

Problem 3:

Given a string containing just the characters '(' and ')', return the length of the longest valid (well-formed) parentheses substring.

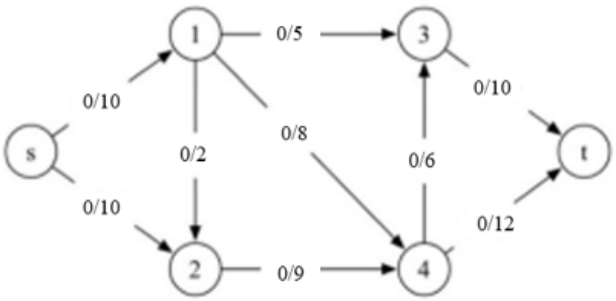
For example, if the input is ")()())", the output is 4, because the longest valid parentheses substring is "()()".

Problem 4:

Suppose there are n players playing games. Every player plays exactly one game with every other player. Every game has a winner (there are no ties). Given an input vector $(v_1, ..., v_n)$, design an efficient algorithm to determine whether it is possible for player i to win exactly v_i games, for $1 \leq i \leq n$.

Problem 5:

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 paths $1 \to 3 \to t$ and $1 \to 4 \to t$, then you should choose $1 \to 3 \to t$, because it is lexicographically smaller than $1 \to 4 \to t$).



Problem 6:

The figure shows a flow network in which an s-t flow has been computed. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge (edges without boxed numbers have no flow being sent on them). Find a minimum s-t cut in the flow network, and say what its capacity is.