CSE 431/531 Analysis of Algorithms
Problem Set 3

Chen Xu

Due Date: July 18, 2023 23:59 PM EST.

Problem 1. (40%)

Now it is the chance for you to share an algorithm problem. We have learned about Greedy algorithm and Divide and Conquer. There are thousands of problems and variants that belong to these two categories out there. Here we ask you to complete the following tasks:

1. Accurately state your problem. The problem should belong to either Greedy or Divide and Conquer or the combination of both. (20%)

2. Write the solution for your own problem. (20%)

Here is what we are looking for:

- (5%) The problem should be accurate. This means that everything is defined mathematically. The problem does not need to be long.

- (8%) We want to be as creative as possible. For this problem you are allowed to look up the internet. Of course, the instructor will also look up the internet to judge if your problem is novel. Bonus points (2%) if your problem is based on real life situation. Unanswered stackexchange and mathoverflow posts etc. still count as novel. We would suggest starting from a well known problem and make alternations. But if the alternation makes no significant difference to the original solution then you will lose some points.

- (7%) The problem should be challenging. We will test if AI can answer your problems. You may test yourself as well. The instructor will pick the top 5 most interesting ones to give bonus (5%).

- (20%) The solution must be complete. This includes correctness, time and space complexity.

You have the copyrights of your own problem. It is possible that some variants of your problem may appear as a question in the final exam random question pool. In your response, please indicate if you would like to share with the class and authorize the instructor to change and put your problem in the final exam. If so, then you need to write your response in .tex/.doc.

Problem 2. (30%)

Design a dynamic programming algorithm to solve the subword counting problem of Homework 2. State the recursive structure of your algorithm and why it works. Analyze the running time.

Problem 3. (30%)

Given a $n \times n$ 2D grid of positive integer numbers. You place yourself at the top left corner. The bottom right corner is your destination. You can only move right or down. Your score is the sum of the numbers you get along the way. Design an algorithm to compute the path you take that maximizes this score.