CMPSC 565: Design and Analysis of Algorithm **(Due 2024-10-11 10:00pm)**

Problem Set 2

Instructor: Dr. Antonio Blanca TAs: Qiyu Chen, Jianqiang Li

Notice: Type your answers using LaTeX and make sure to upload the answer file on Gradescope before the deadline. Recall that for any problem or part of a problem, you can use the "I'll take 20%" option. For more details and the instructions read the syllabus.

Problem 1. Longest Increasing Subsequence Problem

Given a sequence of integers $X = \{a_1, a_2, \ldots, a_n\}$, provide an $O(n \log n)$ time algorithm to find the longest increasing subsequence of X.

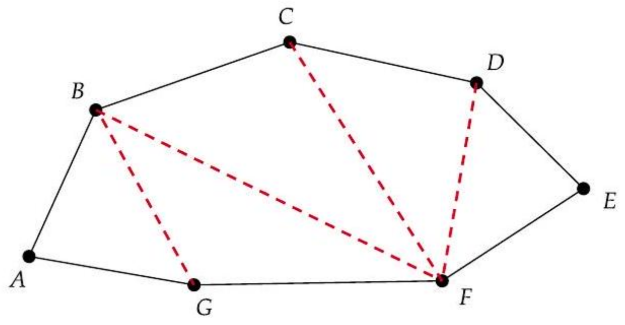
Problem 2. Line Fill

Consider a typesetting software (such as $\TeX$) that converts a paragraph of text into PDF documents (say). The input text is a sequence of n words of width $w_1, \ldots, w_n$. The software needs to somehow minimize the amount of extra spaces, as follows. Every line has width W, and if it contains words i through j, then amount of extra spaces on this line is $W - j + i - \sum_{i \le k \le j} w_k$, because we leave one unit of space between words. The amount of extra space on each line must be nonnegative to avoid overflowing of words. Our goal is to minimize the sum of squares of extra spaces on all lines except the last. Give a dynamic programming algorithm for this problem that runs in $O(n^2)$. Correctness proof is optional for this question.

Problem 3. Short Poly

You have a wood plank in a shape of a convex polygon with n edges, represented as the (circular) list of the 2D coordinates of its n vertices. You want to cut it into $(n-2)$ triangles using $(n-3)$ non-crossing diagonals, such that the total length of the cutting trace (i.e., total length of the picked diagonals) is minimized. Design a dynamic programming algorithm runs in $O\left(n^3\right)$ time to solve this problem. Your solution should include definition of subproblems, a recursion, how to calculate the minimized total length (i.e., the termination step), and an explanation of why your algorithm runs in $O\left(n^3\right)$ time.

For example, in the convex polygon given below, one possible way to cut it into triangles is illustrated with red dashed lines, and in this case the total length of cutting trace is $\overline{BG} + \overline{BF} + \overline{CF} + \overline{DF}$.



Problem 4. Card Color Guessing

Assume a deck of cards has x red cards and y black cards. Alice plays a game, where she chooses a card uniformly at random and guesses if it is red or black. The card is observed and discarded.

Alice does this for $t \le x + y$ rounds always guessing whichever color appears more often in the remainder of the deck (and guessing both with equal probability if there is no difference).

1. Design an $O(x \cdot y)$ time algorithm that computes the probability that after drawing $(x+y)-(i+j)$ cards, there are exactly i red and j black cards left in the deck.

2. Design an $O(x \cdot y)$ time algorithm that computes the expected number of correct guesses of Alice (after all cards are drawn).

Problem 5. Push Shapes

There are n dominos of height h located at positions $x_1, \ldots, x_n$ on a line. All the positions are distinct. When a domino falls down, it either falls left with probability p or to the right with probability $1-p$. If the domino hits another domino in its way down, that domino will fall in the same direction as the domino that hit it. A domino can hit another domino if and only if the distance between them is strictly less than h. For instance, suppose there are 4 dominos located at positions 1, 3, 5 and 8 and $h = 3$. The domino at position 1 falls right. It hits the domino at position 3 and it starts to fall too. In it's turn it hits the domino at position 5 and it also starts to fall. The distance between 8 and 5 is exactly 3, so the domino at position 8 will not fall.

While there are dominos standing, a kid will select either the leftmost standing domino with probability q or the rightmost standing domino with probability $1-q$ and will make it fall (it will fall to the left with probability p or to the right with probability $1-p$). We would like to know the expected total length of the line covered with fallen dominos after all of them are pushed over.

Write a DP algorithm to solve this problem. The running time should be $O(n^2)$. Hint: make sure to work out a few small examples first.

Problem 6. Optimal Balanced Teams

You are organizing a Capture the Flag game on campus. To facilitate this, it is necessary to form two balanced teams from the interested players. The challenge arises from the fact that groups of friends only want to participate together, and you want to include the most amount of people possible. However, you know that to be fair you need the teams to be balanced.

Formally, you are presented with a set of n groups, each containing a_i members and possessing a skill level s_i. The objective is to select two distinct subsets of groups, A and B, such that $\sum_{x\in A} s_x = \sum_{x\in B} s_x$, and $\sum_{x\in A\cup B} a_x$ is maximized. Write a DP algorithm to solve this problem. The running time should be $O(n \times (\sum s_i)^2)$.

Hint: This is a variant of the knapsack problem, where items can't be reused, there are two knapsacks which would have to have the same weight of items.

Coding problems.

Instruction for coding problems: This problem set includes coding problems from `https://codeforces.com`. If you don't have an account, create one. You are asked to implement a solution to each of the assigned problems that passes the systems tests. For each question, you will submit a screenshot showing that your solution has been accepted by the system as correct. You will also include your codeforces handle so that we can check it. The same collaboration policy and guidelines for the homeworks apply to coding problems: while you can discuss and/or consult sources, you are required to write your own code and cite every reference. Instructors may ask students to explain their solution and code.

Problem 7. Multiplicity

`https://codeforces.com/problemset/problem/1061/C`

Problem 8. Substring

`https://codeforces.com/problemset/problem/919/D`

Problem 9. Travel Card

`https://codeforces.com/problemset/problem/756/B`

Problem 10. Soldier and Number Game

https://codeforces.com/problemset/problem/546/D

Coding problems (Extra Credit).

Problem 11. Stars Drawing

https://codeforces.com/problemset/problem/1015/E1

Problem 12. Mahmound and a Message

https://codeforces.com/problemset/problem/766/C

Problem 13. Coloring Trees

https://codeforces.com/problemset/problem/711/C

Problem 14. Flag

https://codeforces.com/problemset/problem/1181/C