CS 3510 – Assignment 3

Due `Friday, June 17, 2022` at `11:59pm` on Canvas

Instructor: Shahrokh Shahi

- Please type your answers (LaTeX is highly recommended) and upload a single PDF file named `<Your-GT-Account>.pdf`, e.g., GBurdell3.pdf, including all your answers. You can submit multiple times. Canvas keeps track of the submissions and append a version number when you re-submit. We always grade your most recent submissions.
- Please read the policies, and do not forget to acknowledge your collaborators and cite your references.
- If you do not understand a question, please ask on Piazza or come to office hours well ahead of the due date.
- It is recommended to start reviewing the course material by reading the lecture slides and reviewing the demo codes. Then, the suggested readings from textbooks and solving the practice problems can provide the additional preparation for solving the homework problems. Please note, for the textbook readings, you do not need to cover the topics which have not been covered in the lectures.

Suggested Reading

	CLRS	KT
Section(s)	Chapter 15	Chapter 6

Suggested Practice Problems

	CLRS	KT
Practice problems	Exercise: 15.1-3, 15.1-4 Problems: 15-2, 15-3	Solved Exercise: 1 Exercise: 1-10, 19-21

Additional reading and problems:
– DPV (Chapter 6)

1 Dynamic Programming: Binary Board [25 pts]

Given a binary matrix B of size $n \times m$, where the entries are either 0 or 1, design a dynamic programming algorithm to find the maximum width w of a square of ones in B, as well as the coordinates (x, y) of the top left corner of such a square. Therefore, for all i and j such that $x \leq i < x + w$ and $y \leq j < y + w$, we have $B[i, j] = 1$.

a) (10 pts) Design a dynamic programming algorithm to find the maximum width w and the corresponding top left coordinate (x, y). (Provide the recurrence relation including the base case(s)).

b) (10 pts) Write a pseudocode presenting your algorithm.

c) (5 pts) Analyze the time and space complexity of your algorithm.

(Hint: For solving this problem, you may consider $OPT[i, j]$ as the width of the largest square of ones whose top left corner is $B[i, j]$.)

2 Dynamic Programming: Atlanta MARTA [25 pts]

The Metropolitan Atlanta Rapid Transit Authority (MARTA) is the principal public transport operator in the Atlanta metropolitan area. It was Formed in 1971 as strictly a bus system, and today, it is transporting almost 450,000 passengers a day (bus and train). Currently, MARTA Passes are the cheapest option for those who regularly use MARTA for transportation. Assume the MARTA Passes are sold in three following forms:

- Daily: A 1-day pass sold for $tickets[0]$ dollars;
- Weekly: A 7-day pass sold for $tickets[1]$ dollars;
- Monthly: A 30-day pass sold for $tickets[2]$ dollars.

For instance, $tickets = 2, 7, 20$ means we need to pay \$2, \$7, and \$20 for each daily, weekly,and monthly pass, respectively. The passes allow consecutive days of travel. For example, if we get a weekly pass on day 5, then we can travel for 7 consecutive days which are: day 5, 6, 7, 8, 9,10, and 11. George P. Burdell is a student at Georgia Tech and he has already organized his commuting plan for the upcoming year. In his plan, each day of year is specified by an integer identification number from 1 to 365. Therefore, he can represent his commuting plan as an array of integers. For instance, days = 8, 9, 10, 11, 14, 17, 18, ... means George needs to commute to the school on the 8th, 9th, 10th, 11th, ... days of year. He asked you to help him find the minimum amount of money that he should spend to purchase MARTA Passes for commuting to school in the next year.

a) (10 pts) Explain the optimal substructure of this problem, and write a recursive expression for calculating min Cost including the base case.

b) (10 pts) Give the pseudocode of a linear Dynamic Programming algorithm to return the minimum cost of commuting every day in the array "days", if the cost of MARTA passes is given in a three-element array "tickets".

c) (5 pts) Analyze the time and space complexity of your algorithm.