CSE 101 Fall 2018
Homework 7
Due: Thursday, December 6, 2018 at 11:59pm

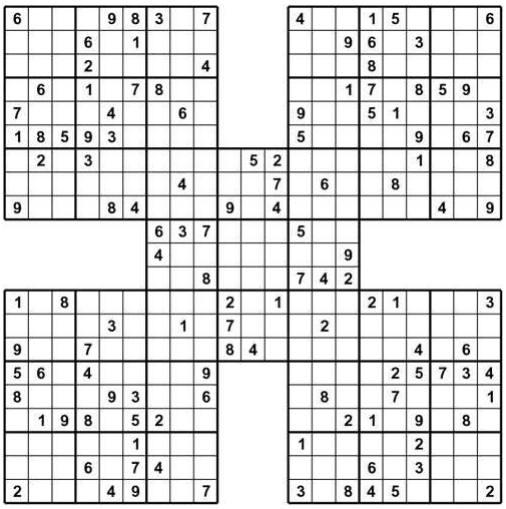
INSTRUCTIONS

This homework assignment may be completed in groups of size 1-4. The solutions must be **typed** (using a computer). Figures and graphs can be handdrawn. For algorithm descriptions we require a high-level English description AND an implementation description. If you find it necessary to also include a pseudocode to help understand your description then feel free to include it.

Please refer to the course page for requirements in writing up answers for algorithm questions.

---

1. (25 points) In usual sudoku, given a partially filled 9x9 2D array 'grid[9][9]', the goal is to assign digits (from 1 to 9) to the empty cells so that every row, column, and subgrid of size 3x3 contains exactly one instance of the digits from 1 to 9. In Samurai Sudoku, 5 sudoku puzzles are merged as shown below. The rules are the same for each of the sudoku puzzles, but the common corner grids have to be compliant with the rules of both the sudoku puzzles they are common to.



Give an algorithm to solve a Samurai Sudoku Puzzle taking hints from what was taught in class.

Give the problem instance, solution format, constraints, and objective for the problem (5 points). Give a high level algorithm (15 points). Give the implementation details for handling such a puzzle (2 points). What is your runtime for this algorithm (3 points)?

2. (25 points) Minimizing the cost of trip from San Diego to Seattle. You are traveling from San Diego to Seattle by car and you need to have three stopovers on the way for your car to be re-filled with gas. You are given:

   (a) a number of choices of towns for each stop.

   (b) a number of hotels to choose from in each city.

   Each trip has a different distance resulting in a different cost (gas). Hotels in these cities have different costs. The goal is to select a route to a hotel in Seattle so that the overall cost of the trip is minimized.

For example, from San Diego you may go to Los Angeles, San Francisco, or San Jose and each has a different cost of travel from San Diego:

San Diego to Los Angeles: 60
San Diego to San Francisco: 255
San Diego to San Jose: 235

The next day, you travel to another city, for example:

Los Angeles to Sacramento: 195
San Francisco to Eugene: 265
etc.

In addition, each of these cities has many hotels 1,2,3,4,...,k that you can stay at with different costs, given by an array $HotelCost_{city}$ at each city. If you make a stopover at a city, you will need to stay there for the night at one of the hotels.

Identify the sub-problems you need to solve to solve the bigger problem (5 points). Give a recursion among sub-problems that you would use to solve this problem (10 points). Give a high level description of your algorithm (10 points).

3. (25 points) A library has a collection of n books that must be stored in alphabetical order on adjustable height shelves. Each book has a height h and a thickness t. The width of the shelf is fixed at W, and the sum of the thicknesses of books on a single shelf must be at most W. The height of each shelf is equal to the height of maximum height book on that shelf. Give an algorithm that minimizes the total height of shelves used to store all the books. You are given the list of books $b_1, b_2, \ldots, b_n$ in alphabetical order and you must organize the books in that order. The i-th book $b_i = (h_i, t_i)$, where h_i is its height and t_i is its thickness.

   Give a backtracking approach (10 points) and then convert it into a dynamic programming approach (10 points). What is your worst case run time for the dynamic programming approach (5 points)?

4. (25 points) Given two strings str1 and str2, and the below operations that can be performed on str1, find the minimum number of edits (operations) required to convert str1 into str2.

   Insert
   Remove
   Replace
   All of the above operations are of equal cost.

   For example: you can convert **caution** to **carton** by deleting an i and replacing a u with an r. As well as by deleting an i and a u and then inserting an r. But you would obviously prefer the first one because it has 2 operations instead of 3 and hence minimizes the cost.

   Identify the sub-problems (5 points). Give a high level description of a backtracking algorithm using this recursion and briefly describe how you would convert this into a dynamic programming algorithm (15 points). What is the run time of your algorithm before applying the dynamic programming approach (5 points).