Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement of Computer Science
Markus Püschel, David Steurer

19. November 2018

Algorithms & Data Structures — Homework 9 — HS 18

Exercise Class (Room & TA): _____

Submitted by: _____

Peer Feedback by: _____

Points: ____

Hint: This exercise sheet is concerned with *dynamic programming*. A complete description of a dynamic program always includes the following aspects (important also for the exam!):

1. **Definition of the DP table:** What are the dimensions of the dynamic programming table DP[., .]? What is the meaning of each entry (in clearly worded words)?
2. **Calculation of an entry:** Which values of the table are initialized, and how are they initialized? How are entries calculated from other entries? What are the dependencies between entries?
3. **Calculation order:** In what order can you calculate the entries so that these dependencies are fulfilled?
4. **Reading the solution:** How can the solution be read out from the table at the end?

Exercise 9.1 *Longest Ascending Subsequence.*

The longest ascending subsequence problem is concerned with finding a longest subsequence of a given array A of length n such that the subsequence is sorted in ascending order. The subsequence does not have to be contiguous and it may not be unique. For example if $A = [1, 5, 4, 2, 8]$, a longest ascending subsequence is $1, 5, 8$. Other solutions are $1, 4, 8$, and $1, 2, 8$.

Given is the array:

$$[19, 3, 7, 1, 4, 15, 18, 16, 14, 6, 5, 10, 12, 19, 13, 17, 20, 8, 14, 11]$$

Use the dynamic programming algorithm as described in class or the script to find the length of a longest ascending subsequence and the subsequence itself. Show all necessary tables and information you used to obtain the solution.

Exercise 9.2 *Longest Common Subsequence.*

Given are two arrays, A of length n, and B of length m, we want to find the their longest common subsequence and its length. The subsequence does not have to be contiguous. For example, if $A = [1, 8, 5, 2, 3, 4]$ and $B = [8, 2, 5, 1, 9, 3]$, a longest common subsequence is $8, 5, 3$ and its length is 3. Notice that $8, 2, 3$ is another longest common subsequence.

Given are the two arrays:

$$A = [7, 6, 3, 2, 8, 4, 5, 1]$$

and

$$B = [3, 9, 10, 8, 7, 1, 2, 6, 4, 5],$$

Use the dynamic programming algorithm as described in class or the script to find the length of a longest common subsequence and the subsequence itself. Show all necessary tables and information you used to obtain the solution.

Exercise 9.3 *Gym Schedule* **(1 Point)**.

Alice likes to lift weights at the gym and wants to schedule her gym going for the next N days. Each day, Alice either goes to the gym or not, so a schedule can be thought of as a list of length N where the i^{th} entry contains whether or not Alice goes to the gym that day. Alice cannot go to the gym on two consecutive days, or else she will become too fatigued.

Use dynamic programming to help Alice calculate the number of different gym schedules under this constraint. Note that one valid schedule is that Alice will not go to the gym at all during those N days.

Exercise 9.4 *Black and White Stones* **(2 Points)**.

Two friends named Tim and Gordon play a game. They takes turns drawing stones from a bag. The bag contains black stones and white stones. Whoever draws the final black stone wins the game. The bag is opaque and the stones are indistinguishable by touch—thus, Tim and Gordon draw stones from the bag randomly without knowing their color in advance. Tim always draws first. The bag is guaranteed to contain at least one black stone.

Describe a dynamic program to determine the probability that Tim will win the game, if it is played with m black stones and n white stones.

Submission: On Monday, 26.11.2018, hand in your solution to your TA *before* the exercise class starts.