CSCI 3104 Spring 2023
Instructors: Ryan Layer and Chandra Kanth Nagesh

Homework 24

Due Date .... April 13, 2023
Name .... **Blake Raphael**
Student ID **109752312**
Collaborators .... **Alex Barry, Brody Cyphers, and Ben Kohav**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.

- You should submit your work through the **class Gradescope page** only (linked from Canvas). Please submit one PDF file, compiled using this LaTeX template.

- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).

- You are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.

- Posting to **any** service including, but not limited to Chegg, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.

- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (I agree to the above, Blake Raphael). □

3 Standard 24: Backtracking to find Solutions

3.1 Problem 1 (2 Points)

Problem 1. Consider the Knapsack problem, with input $[(2,1),(3,2),(1,3),(5,4),(4,5)], W = 10$. Here, each pair (v_i, w_i) denotes an item with value v_i and weight w_i. The following is the dynamic programming table for the optimum value. From the table, **clearly indicate the steps you take to backtrace to find the optimum knapsack.** Be sure to include the steps at which an item was *not* chosen, as well as those at which it was chosen. That is, the number of steps your algorithm takes should be the same as the length of the list (5) (or maybe 6, depending on how you handle the last step).

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
(2, 1) 1	0	2	2	2	2	2	2	2	2	2	2
(3, 2) 2	0	2	3	5	5	5	5	5	5	5	5
(1, 3) 3	0	2	3	5	5	5	6	6	6	6	6
(5, 4) 4	0	2	3	5	5	7	8	10	10	10	11
(4, 5) 5	0	2	3	5	5	7	8	10	10	10	11

Answer. Step 1: $KP[5,10] = max\{KP[4,10], 4 + KP[4,5]\} = 11$. Item 5 is not considered.

Step 2: $KP[4,10] = max\{KP[3,10], 5 + KP[3,6]\} = 11$. Item 4 is considered.

Step 3: $KP[3,6] = max\{KP[2,6], 1 + KP[2,3]\} = 6$. Item 3 is considered.

Step 4: $KP[2,3] = max\{KP[1,3], 2 + KP[1,1]\} = 5$. Item 2 is considered.

Step 5: $KP[1,1] = max\{KP[0,1], 2 + KP[0,0]\} = 2$. Item 1 is considered.

Step 6: $KP[0,0] = 0$. Item 0 is considered.

□

3.2 Problem 2 (2 Points)

Problem 2. Consider the sequence alignment problem where the cost of *sub* and the cost of *indel* are all 1. Given the following table of optimal cost of aligning the strings AACTGA and AAACTG, draw the backward path consisting of backward edges to find the minimal-cost set of edit operations that transforms AACTGA to AAACTG. Besides indicating the backward path, you must also give the minimal-cost set of edit operations. Clearly explain your work.

		A	A	A	C	T	G
	0	1	2	3	4	5	6
A	1	0	1	2	3	4	5
A	2	1	0	1	2	3	4
C	3	2	1	1	1	2	3
T	4	3	2	2	2	1	2
G	5	4	3	3	3	2	1
A	6	5	4	3	4	3	2

Answer. Our backwards path will be the following:

$(6,6)$ insert, delete → $(6,5)$ insert → $(5,4)$ insert → $(4,3)$ insert → $(3,2)$ insert → $(2,2)$ no-op → $(1,1)$ no-op → $(0,0)$ no-op

In order to get to AAACTG from AACTGA, the minimum operations are an insert of A between the second A and C and a delete of the A on the end of the string shown by $(6,6)$.
In order to get to AAACTG from AACTG, we need to insert an A between the second A and C, shown by $(6,5)$.
In order to get to AAACT from AACT, we need to insert an A between the second A and the C, shown by $(5,4)$.
In order to get to AAAC from AAC, we need to insert an A between the second A and the C, shown by $(4,3)$.
In order to get to AAA from AA, we need to insert a third A, shown by $(3,2)$.
In order to get to AA from AA, we need a no-op, shown by $(2,2)$.
In order to get to A from A, we need a no-op, shown by $(1,1)$.
In order to get from an empty string to an empty string, we need a no-op, shown by $(0,0)$.

□