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

Homework 23

Due Date .... Apring 6, 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)

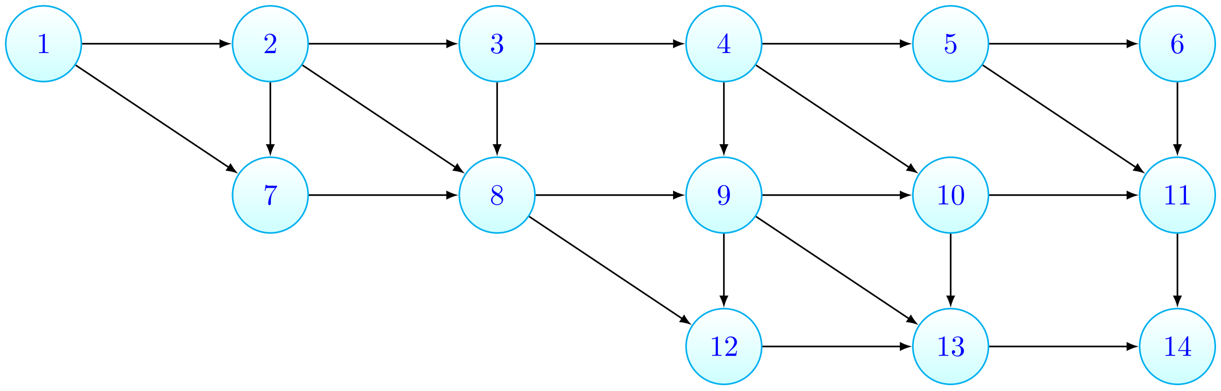
- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided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

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

3 Standard 23- Dynamic Programming: Using Recurrences to Solve

3.1 Problem 1 (2 Points)

Problem 1. Given the following directed acyclic graph. Use dynamic programming to fill in a **one-dimensional** lookup table that counts number of paths from each node j to 14, for $j \geq 1$. Note that a single vertex is considered a path of length 0. **Fill in the lookup table for all vertices 1-14; and in addition, clearly show work for vertices 9-14**.



Answer. Here is the look-up table:

Node j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Number of Paths p	28	23	13	8	2	1	5	5	4	2	1	1	1	0

Here is the work for nodes $9 - 14$:

9:
$9 \rightarrow 10 \rightarrow 11 \rightarrow 14$
$9 \rightarrow 10 \rightarrow 13 \rightarrow 14$
$9 \rightarrow 12 \rightarrow 13 \rightarrow 14$
$9 \rightarrow 13 \rightarrow 14$

10:
$10 \rightarrow 11 \rightarrow 14$
$10 \rightarrow 13 \rightarrow 14$

11:
$11 \rightarrow 14$

12:
$12 \rightarrow 13 \rightarrow 14$

13:
$13 \rightarrow 14$

14:
14

□

3.2 Problem 2 (2 Points)

Problem 2. Consider the following input for the Knapsack problem with a capacity $W = 12$:

item i	1	2	3	4	5	6
value v_i	2	7	18	23	29	35
weight w_i	1	2	5	6	7	9

Fill in a lookup table, similar to the example on page 12 of course notes for week 9 (see Week 9 under "Modules" of the course canvas). In addition, clearly explain how you obtain the maximum values/profits $OPT(6, w)$, $w = 7,\ 8,\ 9,\ 10,\ 11,\ 12$.

Answer. Here is the lookup table where weight is the top row and item is the left-most column:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	2	2	2	2	2	2	2	2	2	2	2
2	0	2	7	9	9	9	9	9	9	9	9	9	9
3	0	2	7	9	9	18	20	25	27	27	27	27	27
4	0	2	7	9	9	18	23	25	30	32	32	41	43
5	0	2	7	9	9	18	23	29	31	36	38	41	47
6	0	2	7	9	9	18	23	29	31	36	38	42	47

Each sub-problem chooses the maximum profit that can be fit within the weight. So for example, when we are calculating the maximum profit for a knapsack weight of 9, we find that item 6 fits at a weight of 9 and items 5 and 2 also fit at a weight of 9. Item 6 by itself is valued at 35 however, items 5 and 2 are valued at 36. We take the maximum value of the items that fit in the knapsack at the sub-problem's weight, so in our example that would be items 5 and 2, hence we get the value 36 in our lookup table.

□