CSCI 3104 PS9a

Jonathan Phouminh

TOTAL POINTS

5 / 12

QUESTION 1

7 pts

1.1 2 / 4

- **+ 2 pts** Correct table
- **+ 2 pts** Correct backward path
- ✓ **+ 1 pts Partially correct table with some errors.**
- ✓ **+ 1 pts Partially correct backward path**
- **+ 0 pts** Table had significant errors.

1.2 1 / 1

- ✓ **+ 1 pts Correct**
- **+ 1 pts** Full credit, but note that the cell above the bottom right cell corresponds to the sub-problem where we can choose elements from { (4, 3), (1, 2), (3, 1), (5, 4) } and the knapsack has a weight value of 10. You really want to look at the bottom-right cell.
- **+ 0 pts** Incorrect or no solution
- **+ 0.5 pts** Correct optimal value, but without optimal cell
- **+ 0.5 pts** Correct optimal cell without optimal value

1.3 2 / 2

- ✓ **+ 2 pts Correct**
- **+ 0 pts** Incorrect or no solution
- **+ 1.5 pts** Missing one item or Missing weight-value values (total)
- **+ 1 pts** Correct weight and value, but incorrect items or did not list items

QUESTION 2

2 0 / 5

- **+ 5 pts** Correct
- **+ 4 pts** Minor errors
- ✓ **+ 0 pts No or incorrect solution**
- **+ 3 pts** Several errors
- **+ 1 pts** Significant errors
- **+ 0 pts** You have to write the count of paths from each node to 14.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 9a (12 points) **Fall 2019, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Gradescope if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 9a (12 points) **Fall 2019, CU-Boulder**

1. (7 pts) Consider the following DP table for the Knapsack problem for the list $A = [(4,3),(1,2),(3,1),(5,4),(6,3)]$ of (weight, value) pairs. The weight threshold $W = 10$.

    - Fill in the values of the table.
    - Draw the backward path consisting of backward edges and do not draw (or erase them) the edges that are not part of the optimal backward paths.

    (a) (4 pts) Fill the table with the above requirements (You can also re-create this table in excel/sheet).
    Solution.

			index for j										
value	weight	index for i	0	1	2	3	4	5	6	7	8	9	10
2	1	0	0	2	2	2	2	2	2	2	2	2	2
1	3	1	0	2	2	2	3	3	3	3	3	3	3
3	4	2	0	2	2	2	3	5	5	5	6	6	6
4	5	3	0	2	2	3	3	5	6	6	6	7	9
3	6	4	0	2	2	3	3	5	6	6	6	7	9

    (b) (1 pts) Which cell has the optimal value and what is the optimal value for the given problem?
    Solution.

    For cells, F(i,j)
    The cell that holds the optimal value is F(4,10), or the cell in the last row and last column.

    (c) (2 pts) List out the optimal subset and provide it's weight and value.
    Solution.

```
For items = I(weight,value)

Optimal subset = {I(5,4),I(4,3),I(1,2)}
Total Weight = 10
Value = 9
```

Name: Jonathan Phouminh

ID: 106054641

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**

Problem Set 9a (12 points) **Fall 2019, CU-Boulder**

2. (5 pts) Given the following directed acyclic graph, and assume a "path" must have at least one edge in it to be well defined. Use dynamic programming to fill in a table that counts number of paths from each node j to 14, for $j \geq 1$.

`dag_ps9.pdf`

Solution.