

Name:

ID:

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
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: Nikolaas Bender

ID: 106977096

CSCI 3104, Algorithms

Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal

Fall 2019, CU-Boulder

1. (11 pts) Solve the following based on the given intervals.

Event Index	Interval	Value
1	[1, 10]	10
2	[11, 20]	5
3	[21, 30]	15
4	[9, 12]	7
5	[19, 22]	8

- (a) (2.5 pts) For the given set of requests with a start and finish time and a value, what is $p(j)$ for each request j ?

Solution.

Consider	Optimal Value	$p(j)$
1	10	0
2	15	1
3	30	2
4	30	3
5	30	4
	config	

Name:
ID:

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

- (b) (1.5 pts) Assume you are given a greedy algorithm for the weighted interval scheduling problem that selects the requests in descending order of their value, i.e. highest valued requests are selected first. Provide an example set of intervals where this approach won't produce the optimal solution.

<i>Solution.</i>	Event Index	Interval	Value
	1	[1, 10]	5
	2	[1, 5]	3
	3	[6, 10]	3

optimal would choose indices 2 and 3 to get a value of 6 however the greedy would choose index 1 and get a value of 5.

Name:
ID:

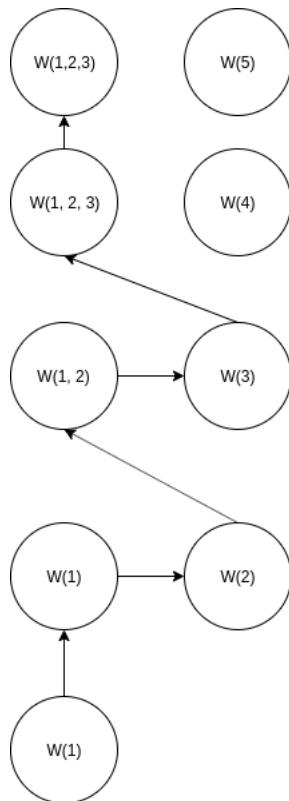
CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

- (c) (2 pts) In the following recursive formula for the weighted interval scheduling problem in Question 1, how many recursive calls are there to $WISNoMem$? Show the recursion tree to support your answer.

```
WISNoMem(j) {  
    if (j == 0)  
        return 0  
    else  
        return max(WISNoMem(j-1), v[j] + WISNoMem(p[j]))  
}
```

Solution. 9 calls to $WISNoMem$



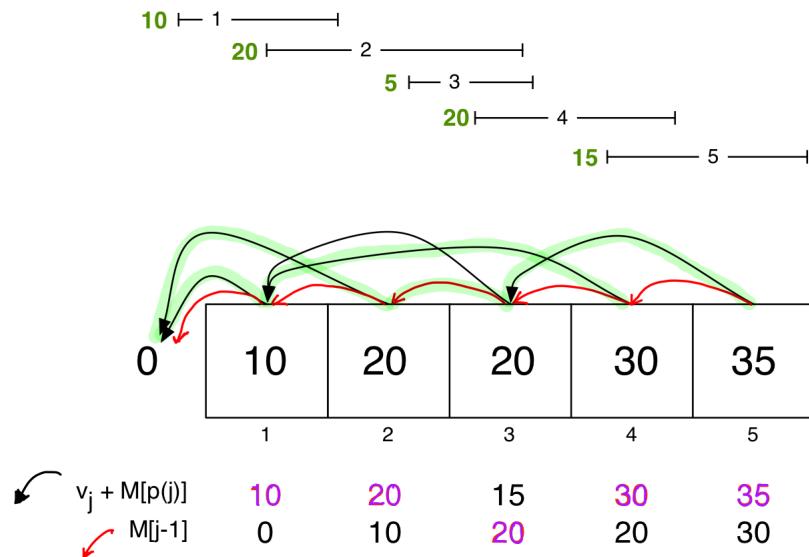
Name: Nikolaas Bender
ID: 106977096

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

- (d) (3 pts) Using the bottom-up iterative algorithm presented in lecture, show the $OPT(j)$ values for all requests. Also draw the arrows to show the previous subproblem you used to fill an entry.

Solution. I don't fully understand what the question is asking as it seems to roughly be the same as the previous questions. However shiv posted this on piazza. Its a trace back of the optimal as you consider different events. I am not completely clear on where some of the numbers came from but with the black and red arrows its showing the max of adding to what exists or to choose a different optimal, this decision goes back to the core of DP.



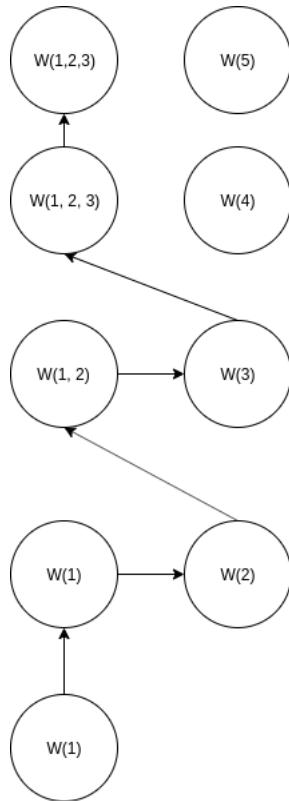
- (e) (2 pts) Show which requests are selected in the optimal solution and the value of the solution. Provide a 2-3 sentence explanation of how you retrieved the solution using the arrows you drew.

Solution. Choosing the right meetings is relatively easy. As a new time is considered we check if the meeting time conflicts and we see if the added value is more than what is already there. Then there are more considerations for possibilities that were passed over earlier but a completely different combination is the optimal than what the algorithm chooses. however that never seems to be the case.

Name:
ID:

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder



2. (14 pts) Solve the following sequence alignment questions.

- (a) (2 pts) Consider the sub-problem represented by i and j i.e. $\text{cost}[i][j]$ which represents the minimum cost of aligning sub-sequences in the source and target respectively.

What are the smaller sub-problems needed to solve this sub-problem?

What operation do each of the smaller sub-problems represent **with respect to the source sequence?**

Solution. The costs of all elements up through $(i - 1, j)$ and $(i, j - 1)$ so that a decision can be made of the total cost at (i, j) because the cost at (i, j) is dependent on the previous optimal solutions at $(i - 1, j)$, $(i, j - 1)$, and $(i - 1, j - 1)$.

Name:

ID:

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

- (b) (4 pts) Consider the source sequence $s = BEAR$, the target sequence $t = BARE$, and the given cost table. **Draw the arrows that represent the optimal alignment** of s and t . You don't need to draw arrows which are not part of this optimal alignment.

(Note that you have to do the back tracing starting from the cell representing the solution to the full problem and traverse to the base case. You can infer the parent from observing the relevant sub-problems.)

$s \setminus t$	-	B	A	R	E
--	0	1	2	3	4
B	1	0	1	2	3
E	2	1	1	2	2
A	3	2	1	2	3
R	4	3	2	1	2

Also, draw the alignment corresponding to the arrows for the given sequences $s = BEAR$ and $t = BARE$ and **give the set of optimal operations** on the $s = BEAR$ which will transform it to $t = BARE$ (using the cost table with relevant arrows you drew earlier).

An example drawn alignment and set of operations for $s = STEP$ and $t = APE$ look like following. (Show the 'no-op' using a colon (:)) to differentiate with a regular sub.)

Drawn alignment = Ops = ['sub', 'sub', 'no-op', 'delete']

S T E P
| | : |
A P E -

Solution. operations are: [no-op, sub, del, sub, ins]

Name: Nikolaas Bender

ID: 106977096

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

	-	B	A	R	E
-	0	1	2	3	4
B	1	0	1	2	3
E	2	1	1	2	2
A	3	2	1	2	3
R	4	3	2	1	2

- (c) (8 pts) Consider the source sequence $s = \text{PLAIN}$, the target sequence $t = \text{PLANE}$, $\text{cost}(\text{insert}) = 2$, $\text{cost}(\text{delete}) = 2$, and $\text{cost}(\text{sub}) = 2$. Your goal is to align (or transform) the source to the target with minimum cost. Align the source sequence to the target sequence using a bottom-up tabular approach like the one shown on Q2 part b.

- Draw the DP table showing the cost of alignment.
- Show the arrows representing the optimal alignment. Note that there are more than one optimal alignment. Show **two** of the optimal alignments in separate tables using arrows.
One such example for different sequences and costs -
- Draw the alignments corresponding to the **two** optimal alignments like you

Name: Nikolaas Bender

ID: 106977096

CSCI 3104, Algorithms
Problem Set 10a (25 points)

Profs. Hoenigman & Agrawal
Fall 2019, CU-Boulder

x/y	-	A	P	E
-	0	1	2	3
S	1	1	2	3
T	2	2	2	3
E	3	3	3	2
P	4	4	3	3

drew in the last part of Q2 part b.

- Provide the cost of the optimal alignment.

Solution.

The image shows a handwritten dynamic programming table for sequence alignment. The rows are labeled with the letters S, P, I, A, N, E and the columns with P, I, A, N, E. The first row and column are crossed out. The second row has a circled 'P' at its start. The third row has a circled 'I'. The fourth row has a circled 'A'. The fifth row has a circled 'N'. The sixth row has a circled 'E'. The seventh row has a circled 'P'. The eighth row has a circled 'I'. The ninth row has a circled 'A'. The tenth row has a circled 'N'. The eleventh row has a circled 'E'. The values in the table are as follows:

	P	I	A	N	E
S	0	2	4	6	8
P	2	0	2	4	6
I	4	2	0	2	4
A	6	4	2	0	2
N	8	6	4	2	4
E	10	8	6	4	2