Lesson 20- Dynamic Programming

This is a preview of the published version of the quiz

Started: Jul 30 at 1:14pm

Quiz Instructions

- Watch the video below. The quiz has additional videos and readings inside of it.
- Joining the Ohyay is optional, if you want help.
- This assignment is an **individual** submission but you may work with your classmates.
- You will submit this **Canvas Quiz** for this assignment.

Watch

Do

Hey students, this is Dr. Bart!

It looks like AlgoTutorBot is mad that you've been helping me. I see he's assigned one of my old quizzes as punishment. What a jerk! We certainly should all be angry at him, and not me.

This "quiz" is actually just an **open-note** assignment. Don't mention it to AlgoTutorBot, but I noticed that it's **not worth any more points than any other assignment** and there's **no time limit besides the lock date**. So please **do not panic**!

When you submit, **the system will not tell you if you got everything right**, so please make sure you are happy with your answers before submitting. You can submit multiple times, up until the lock date, and it will only take your latest submission.

There are readings and videos embedded in the quiz. I strongly recommend you read and watch everything provided. You might also need to google for additional explanations, or seek help from the instructor, TAs, or even a classmate. Definitely good to work together on this one!

Focus on your learning. This is an explanation of a critical topic that shows up all the time in interviews. Heck, it's probably the most common kind of "curveball" interview question. One of your classmates actually had an interview earlier this semester involving a Dynamic Programming problem, in fact! This is why I'm not just going to give you the answers to the quiz. It's worth your time to learn this stuff.

I'll contact you again in a bit. Now that we have Ada and Babbage, I think I have a plan that will let us find out what AlgoTutorBot is up to. And Dynamic Programming will be useful for it! What a coincidence!

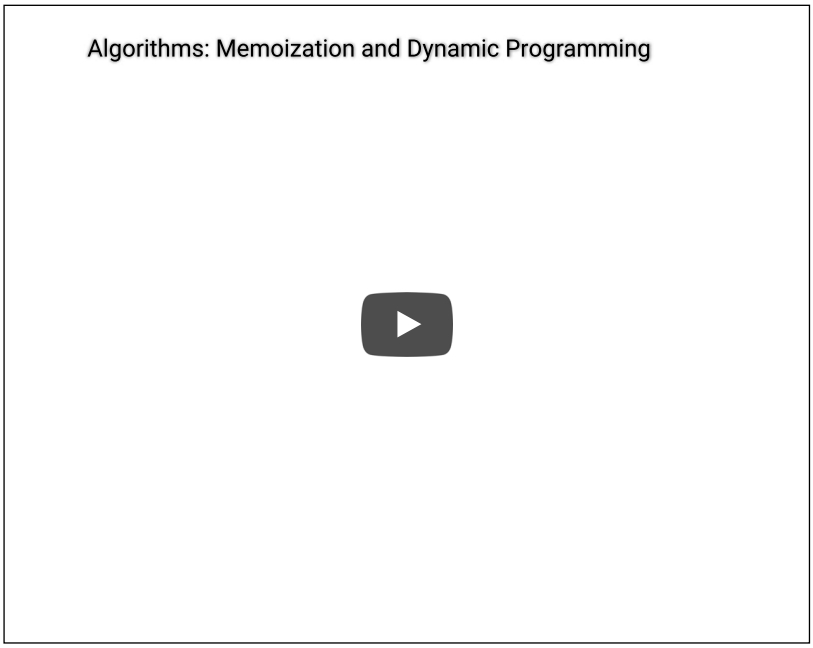
Stay safe, everyone, and good luck with this assignment!

Corrections

So far, there have been no mistakes reported in the quiz. This space will be updated if mistakes are reported.

Begin by reading this page: **All About Dynamic Programming**

Then, watch this video too.



Question 1

1 pts

According to Wikipedia (https://en.wikipedia.org/wiki/Dynamic_programming#History), which of the following is the best description for where the name "Dynamic Programming" comes from.

- ○ The technique involves a programmer dynamically considering many possible aspects.
- ○ The approach uses dynamic memory, as opposed to static memory.
- ○ The approach is actually irrelevantly named compared to what it does.
- ○ The technique can only be used in a dynamic programming language, like Python.
- ○ The approach requires a dynamic array.

Question 2

1 pts

Which is the most accurate statement?

Thanks to dynamic programming...

- ○ Fibonacci is solvable in linear time by storing partial solutions
- ○ Fibonacci is solvable in quadratic time by backtracking through the search space of possibilities.
- ○ Fibonacci is solvable in constant time via a special math formula.
- ○ Fibonacci is solvable in logarithmic time by recursively dividing the search space in half.

Question 3

1 pts

What is the best upper bound for each of the following algorithms?

Naive recursive fibonacci	[Choose]
Memoized fibonacci	[Choose]
Tabulated fibonacci	[Choose]

Question 4

1 pts

When you memoize the fibonacci algorithm in Python, what is an upper bound for the **space** that it requires?

- ◯ Quadratic, because we have repeated subproblems
- ◯ Linear, because we have repeated subproblems
- ◯ Linear, because the size of the call stack and a memoization dictionary,
- ◯ Quadratic, because we have both a call stack and a memoization dictionary
- ◯ Exponential, because we have repeated subproblems
- ◯ Constant, because we can solve the problem "in-place"

Question 5

1 pts

Match the term to its best definition.

Memoization	[Choose]
Tabulation	[Choose]
Dynamic Programming	[Choose]
Divide and Conquer	[Choose]
Greedy	[Choose]

Question 6

1 pts

Indicate which statements are generally true.

- [ ] Any algorithm that is recursive can be optimized by using a Dynamic Programming strategy.
- [ ] Any function can be tabulated by storing its array elements in a table.
- [ ] Any dynamic programming algorithm can be expressed as a recurrence relation.
- [ ] Any function can be memoized by storing its arguments and return values, even if it does not affect its time complexity.

Question 7

1 pts

Which of the following would prevent a problem from being solved with dynamic programming?

- [ ] The input of the problem cannot be stored as a 2D array.
- [ ] The problem requires exponential time.
- [ ] Subproblems do not appear more than once.
- [ ] The problem cannot be divided into smaller problems.

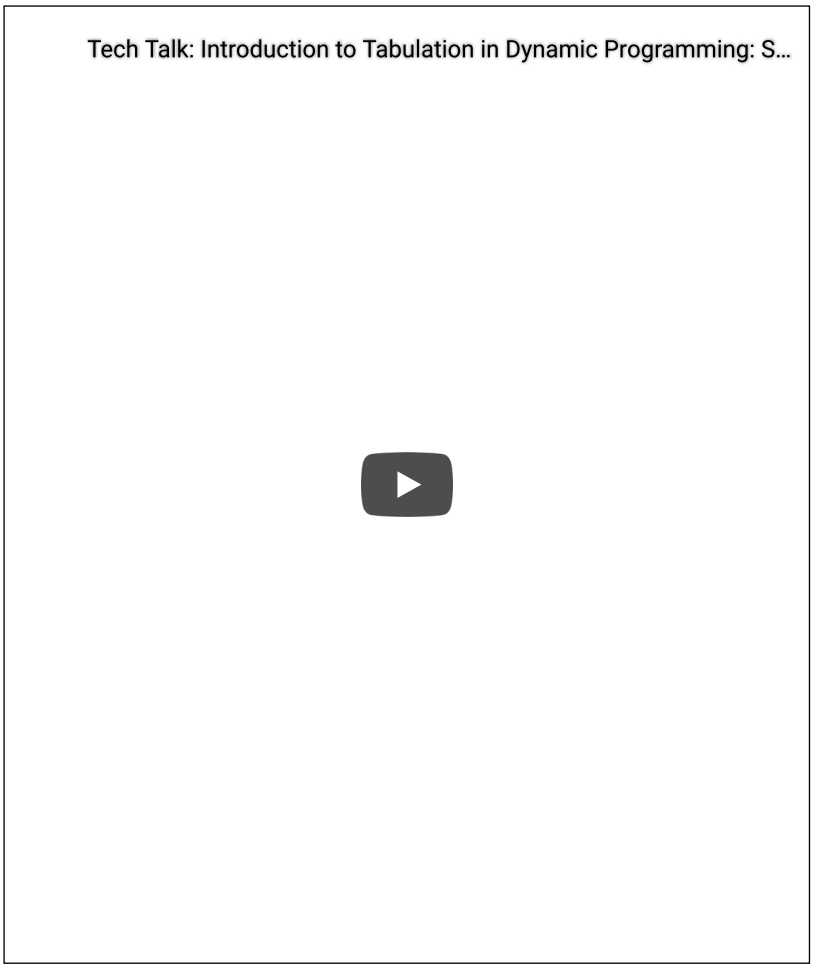
Question 8

1 pts

The table below holds an Edit Distance tabulation for a conversion of the string "babbage" to "ada". Insertions and deletions cost 1, while substitutions are not allowed. Fill in the remaining cells.

		a	d	a
	0	1	2	3
b	1			
a				

b				
b				5
a			5	
g				
e	7			6



Question 9 1 pts

The Subset Sum problem can be solved more efficiently with tabulation than without. Which statements are true?

- [ ] The vertical axis of the table represents the subset of values where the value at left is added to the previous row.
- [ ] A cell of the table contains True if there exists a subset of the values for that row that equal the sum represented by the column.
- [ ] In order to determine if the sum of a set `X` is equal to `S`, it helps to know what values smaller than `S` can be formed from the subsets of `X`.
- [ ] The top row represents the solution for a given desired sum.

- [ ] The top row represents the largest value seen in the subset.
- [ ] The cells of the table contains an integer representing how many solutions there are in a neighboring cell.

Question 10

1 pts

Which of the following describe situations where Dynamic Programming leads to a more efficient solution than alternatives?

You may have to research these algorithms if you do not recall them.

- [ ] Merge sort can solve the Sorting Problem
- [ ] Binary Search can solve the Search Problem
- [ ] Djikstra's Algorithm can solve single-source shortest path
- [ ] Depth-first Search can solve the Connected Graph problem
- [ ] Wagner–Fischer algorithm can solve Edit Distance
- [ ] All algorithms can be solved more efficiently using Dynamic Programming

Quiz saved at 1:14pm Submit Quiz