Lesson 23- Nondeterministic Polynomial Time

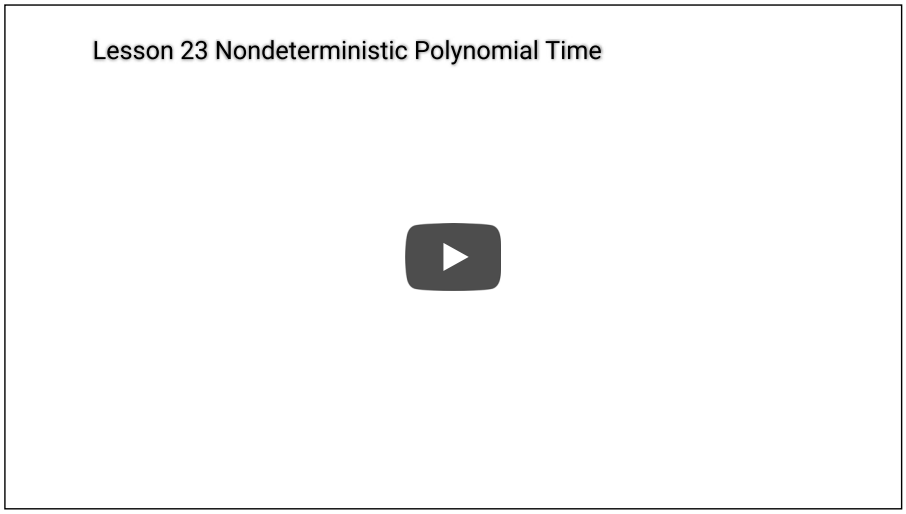
ⓘ This is a preview of the published version of the quiz

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!

Another quiz?? I'm starting to think AlgoTutorBot believes you really enjoy these things. Still, it's important that we take it seriously. The theory stuff can be a little dry, but it's valuable to get the gist of it. Especially if you want to take further Computer Science theory courses! But it won't be too much longer before we can done waiting, and we can set my big plan in motion. So just hang in there!

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.

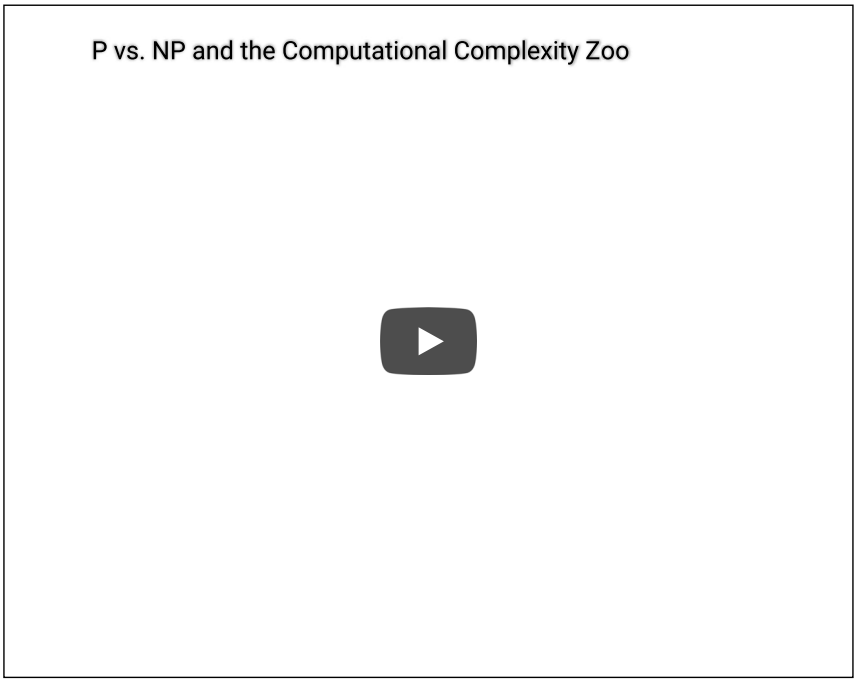
There are readings and videos embedded in the quiz. I strongly recommend you read and watch everything provided. You might also need to google for additional explanations, or seek help from the instructor, TAs, or even a classmate. Definitely good to work together on this one!

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.

P vs. NP is one of the most important problems in Computer Science, and we may never solve it. It's important that you leave this course conversant in it, if only to be able to make stimulating dinner table conversation about it.



Question 1

1 pts

Mark all of the following that are True statements.

- [ ] The term "Non-polynomial" in NP refers to how NP problems cannot be solved in polynomial time, but require exponential time.
- [ ] All problems can be classified as either P or NP: either easy or hard.
- [ ] The term "Non-Deterministic" in NP refers to how you can verify a solution in polynomial time, but you cannot compute the solution in polynomial time without guessing.
- [ ] The question P vs. NP is a question asking the relationship between two sets of problems.

Question 2

1 pts

Mark each statement as True, False, or Unknown.

All of the problems in NP are in NP-Complete	[Choose]
All of the problems in NP are in NP-Hard	[Choose]
All of the problems in P are in NP	[Choose]
All of the problems in P are in NP Complete	[Choose]
All of the problems in NP-Complete are in NP	

[Choose]

All of the problems in NP-Hard are in NP [Choose]

Question 3 1 pts

I recommend you read **this page** (https://eklitzke.org/the-traveling-salesman-problem-is-not-np-complete) to better understand the nuance of NP/NP-Complete/NP-Hard; but don't expect it to just answer this entire question.

Mark all of the following that are True statements.

- [ ] NP Hard problems are too hard to compute, which is why they are always impossible.
- [ ] Small instances of NP-Complete problems are necessarily harder to solve than even large instances of P problems.
- [ ] If a problem is in NP, then it must necessarily require exponential time to compute a solution.
- [ ] NP problems are as hard as it gets - nothing is worse than an NP problem.
- [ ] It is possible to solve an NP-Complete problem optimally in certain cases.
- [ ] P problems are always solvable in an acceptable amount of time (i.e., less than a human timespan), no matter what case is given.
- [ ] For the majority of NP-Complete problems' instances, the time required to compute a solution is too prohibitive.

Question 4 1 pts

Match each definition to its most appropriate term. You may find **this Stack Overflow post** (https://cs.stackexchange.com/questions/75066/what-are-the-differences-between-search-problems-optimization-problems-and-dec) to be helpful.

Find the appropriate output for a given input. [Choose]

Find whether there is a solution to the problem. [Choose]

Find any solution to a problem. [Choose]

Find the best solution to a problem from among all possible solutions. [Choose]

What kind of problem is this question best described as? [Choose]

Question 5 1 pts

You should try to get a list in your head of all the NP-Complete problems, so when you encounter them you know how difficult the problem gets at its worst.

Karp's 21 NP Complete Problems (https://en.wikipedia.org/wiki/Karp%27s_21_NP-complete_problems) is a list of particularly important problems, since they were one of the first groups to be proven NP Complete.

The Algorithm Design Manual offers nice, concise descriptions of each one:

- Clique (https://www.algorist.com/problems/Clique.html)
- Set Packing (https://www.algorist.com/problems/Set_Packing.html)
- Job Sequencing (https://www.algorist.com/problems/Job_Scheduling.html)
- Hamiltonian Cycle (https://www.algorist.com/problems/Hamiltonian_Cycle.html)
- Satisfiability (https://www.algorist.com/problems/Satisfiability.html)
- Vertex Cover (https://www.algorist.com/problems/Vertex_Cover.html)
- Chromatic Number (https://www.algorist.com/problems/Vertex_Coloring.html)

Match each problem description to one of Karp's 21 Problems that it most closely resembles.

We have a finite collection of Lego pieces that can be put together into various creations. We also have a collection of manuals that describe how a subset of the Legos can be combined into a given creation. We want to determine the largest subset of manuals we can choose, such that the creations will not require the same pieces.	[Choose]
Your new video game allows players to customize their outfits. However, some combinations are not possible (e.g., you cannot wear two kinds of pants, striped shirts must never go with solid socks unless you also wear a beret). You wish to determine whether a user's given outfit is a valid combination.	[Choose]
I have a herd of Corgis that need to be pet, so I bring them to my Algorithms class. Each student can only pet one Corgi at a time, and each Corgi takes a different amount of time to pet (depending on the Corgi and student). Since we are running out of time in the semester, I need to get all the Corgis' pet as quickly as possible.	[Choose]
Your new social media website has finally attracted its millionth user, and it is time to enter its sinister second phase. Given a list of its users and who they are friends with, we need to determine the largest subgroup of users who all know each other.	[Choose]
You are shipping a package full of AlgoBots between all of your students. There is just one package, and you need to make sure the package circulates to each student. You also want to minimize the total travel time.	[Choose]
We are trying to set up surveillance cameras all around campus, to keep key points monitored. The cameras can only be placed in strategically hidden spots, but can potentially watch multiple spots. We want to determine the smallest number of cameras that we need to place.	[Choose]

Question 6

1 pts

Classify each problem based on the worst case lower bound needed to compute a solution.

Find the minimum of an unsorted list.	[Choose]
Iterate through an array of strings and determine which one has the character "Q".	[Choose]
Determine if all the rooms in a dungeon can be colored "red" (hard) or "blue" (easy) without any easy rooms next to easy rooms or hard rooms next to hard rooms.	[Choose]
Find the shortest path from one vertex in a weighted graph to another	[Choose]
Find the shortest route connecting all vertices in a weighted graph.	[Choose]
Find the smallest set of rooms in a dungeon that connect to every edge.	[Choose]
Analyze any given program and its input to determine if it ever stops.	[Choose]

Turing Machines are another key concept in Computer Science theory, and you should know roughly what they are.

If you have not taken CISC303 Automata Theory, you should watch this video to learn about Turing Machines: **https://www.youtube.com/watch?v=dNRDvLACg5Q** (https://www.youtube.com/watch?v=dNRDvLACg5Q)

(https://www.youtube.com/watch?v=dNRDvLACg5Q)

Question 7 1 pts

Your laptop is equivalent in power to a Turing Machine - that is, anything a Turing Machine could compute you could compute on your laptop (and vice versa).

- ○ True
- ○ False

Question 8 1 pts

A *Non-deterministic* Turing Machine can be used to efficiently solve NP-Complete problems, by checking all the possible paths that a computation might require in Polynomial time. How does this work?

You should take a look at **this Stack Overflow question** (https://cs.stackexchange.com/questions/80563/how-does-a-nondeterministic-turing-machine-work) about Non-deterministic Turing Machines.

- ◯ Via reduction, we simplify the NP problem path to a P problem path and solve that instead.
- ◯ Via backtracking, we recursively iterate through each path simultaneously.
- ◯ Via dynamic programming, they can avoid checking repeated paths.
- ◯ None of these options, NDTMs do not exist in reality.