NTNU
Kunnskap for en bedre verden

NP and Computational Intractability

TDT4121 Assignment Lecture 02/11 - Assignment 7

> An **intractable problem** is a problem in which the only exact solution is one that takes too many resources (time, memory, etc.). In other words, a problem with no efficient solution.
>
> – *https://computersciencewiki.org/index.php/Computational_intractability*

Chapter 8 - NP and Computational Intractability

> *So far we've developed efficient algorithms for a wide range of problems and have even made some progress on informally categorizing the problems that admit efficient solutions. But although we've often paused to take note of other problems that we don't see how to solve, we haven't yet made any attempt to actually quantify or characterize the range of problems that* ***can't be solved efficiently****.*

Goals 🏁

The student should be able to:

- Understand the difference between **P**, **NP**, **NP-Complete** and **NP-Hard** problems.
- Prove whether a problem is **NP** or **NP-Complete**.
- Understand the basis of *Polynomial-Time Reduction*
- Understand the *Satisfiability Problems*
- Understand the *Traveling Salesman Problem*
- Understand the *Hamiltonian Cycle Problem*

Chapter 8 Goals

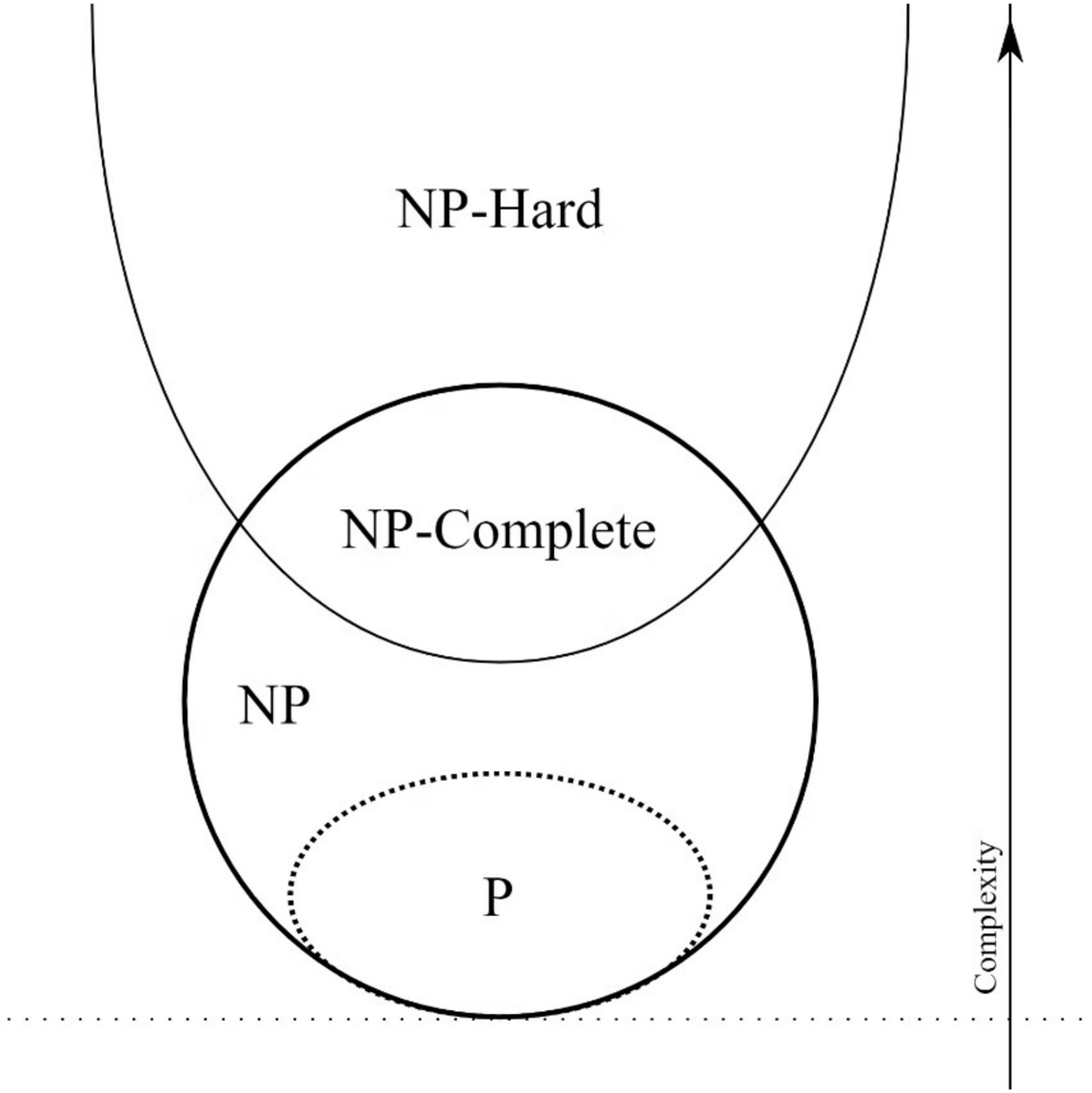
NTNU
Kunnskap for en bedre verden

Contents

Assignment 7 Contents

Complexity Classes

- P : solvable in polynomial time
- NP : yes-solutions verifiable in polynomial time
- NP-Complete : "hardest" problems of NP
- NP-Hard : NP-Complete or harder
- co-NP : no-solutions verifiable in polynomial time



wiki/NP_(complexity)

Task 1 NP and reduction

a) Explain in your own words what characterizes an NP-complete problem?

b) Explain the black box metaphor used in the book

c) Your friend Steve has a problem A that he claims is NP-complete since he can prove that $A \in NP$. Is this sufficient proof that the problem is NP-complete? Explain why or why not.

P

- Most algorithms we have studied in this course
- $T(n) = O(n^k)$

NP

- A superset of P, denoted $P \subseteq NP$
- If the answer is yes, the proof must be verifiable in polynomial time
- Example: Is this sudoku solvable?

co-NP

- A superset of P, denoted $P \subseteq NP$
- If the answer is no, the proof must be verifiable in polynomial time
- Example: Is this sudoku unsolvable?

NP-Complete

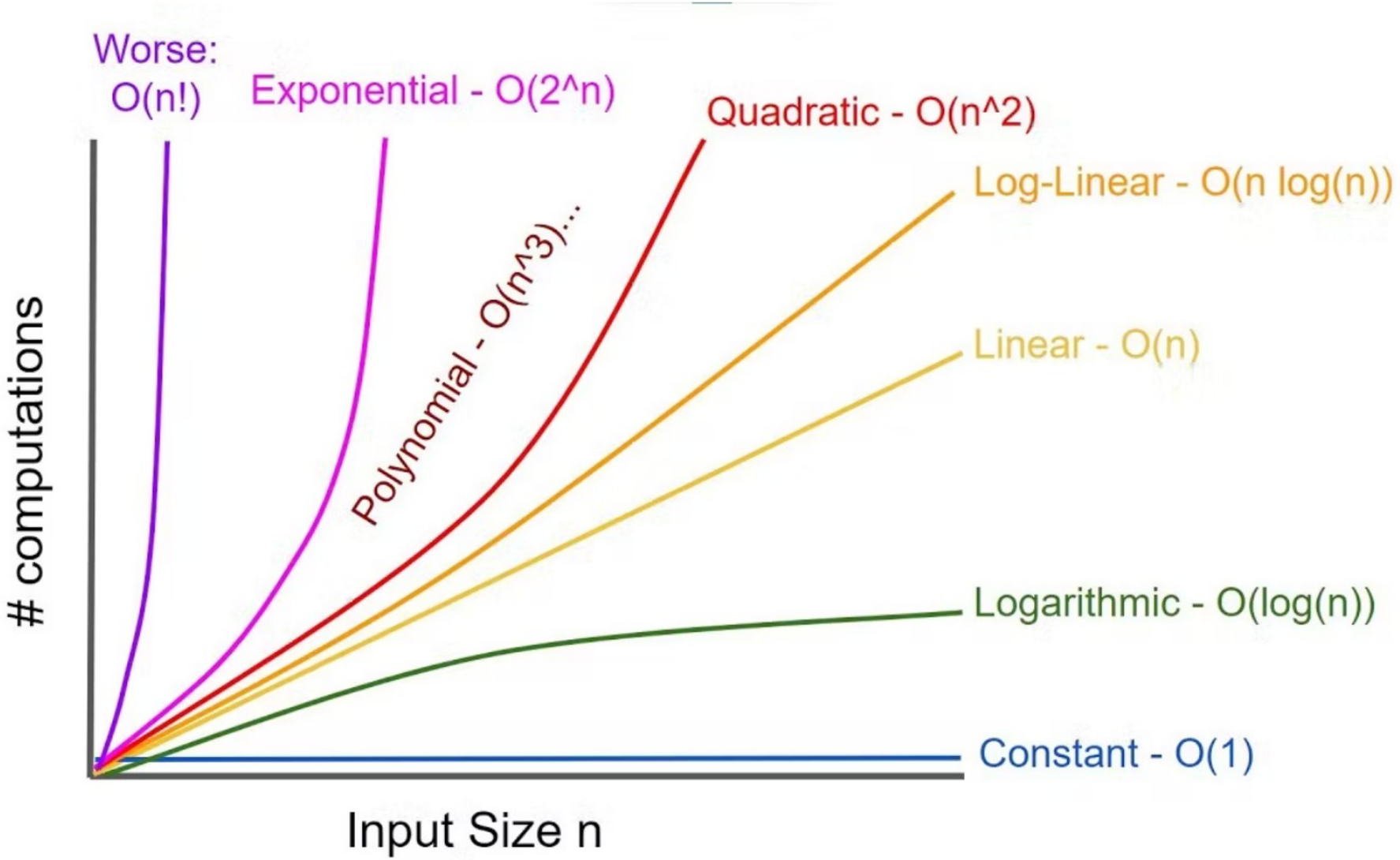
- The "hardest problems" of NP
- Any NP problem can be reduced to an NP-Complete problem
- In other words, if we solve one NP-complete problem efficiently, we can solve them all efficiently
- Example: Does this graph have a Hamilton path?

NP-Hard

- NP-Complete or harder
- Does not have to be verifiable in polynomial time
- Example: Is this the best chess move?

Hardness

- Informally, the harder problem is the one where the optimal algorithm has the largest time complexity.
- More formally, using polynomial time reduction: A problem A is *at least as hard as* a problem B if B is reducible to A. We denote this as $B \leq_p A$



Different Time Complexities

NTNU
Kunnskap for en bedre verden

Why care about NP and Computational Intractability?

Python Showcase

Task 2 NP-complete proof

These problems require that you read and understand the book and recognize the different NP-complete problems and how you can reduce between them

a) Recruitment at NTNU

The recruitment division at NTNU has the following problem. They need at least one professor skilled at each of the c courses at NTNU. They have received applications from p potential professors. For each of the c courses, there is some subset of the p professors qualified in that course. For a given number $k < p$, is it possible to hire at most k professors and have at least one qualified professor in each of the c courses?

We call this problem University Recruitment.

Show that Univerity Recruitment is NP-complete.

b) Campus security

Campus security at NTNU has received a budget cut and has to let go of every security guard except for two: Nancy and Jonathan. They each go out in their car from the same starting location and return to the same location at the end of the day. They want to visit every building on campus. You want to ensure that the route they travel has the minimum possible overlap and that each location is covered by at least one car.

Let's call this problem Efficient Security.

Prove that this problem is NP-complete.

Task 2a, b

List of NP-Complete problems

- Circuit Satisfiability and 3-Satisfiability (3-SAT)
- Vertex Cover and Set Cover
- The Traveling Salesman Problem
- The Hamiltonian Cycle/Path Problems
- The Graph Coloring Problem
- More: Book and https://en.wikipedia.org/wiki/List_of_NP-complete_problems

Polynomial-Time Reduction

- A method for solving one problem using another
- Proves that the first problem is no more difficult than the second one
- Whenever an efficient algorithm exists for the second problem, one exists for the first problem as well
- $X \leq_p Y$

Exercise: Polynomial-Time Reduction

- Let D be the problem of finding a duplicate in a list
- Let S be the problem of sorting a list
- Explain that $D \leq_p S$

If Y is an NP-complete problem, and X is a problem in NP with the property that $Y \leq_p X$, then X is NP-complete.

c) Campus parking

NTNU parking has received budget cuts, so now there is a very limited number of parking spots. Thankfully the staff at NTNU aren't all there at the same time. Meaning two professors can use the same parking spot at different times of the day. The head of parking at NTNU wants to make sure that the parking spots are scheduled so that no two professors park there at the same time.

The problem is the following: Given a parking lot P, can we schedule the staff so that the parking spots do not overlap?

We'll call this problem the campus parking problem.

Prove that this problem is NP-complete.

d) Campus radio

Some of the student unions at NTNU has decided to create their own radio stations with radio shows. The different radio stations are located along the

Task 2c, d

Task 3 The traveling salesman problem

a) **Provide examples of real-world applications for the traveling salesman problem.**

b) **What is a Hamiltonian Cycle, and how is it used in the traveling salesman problem?**

c) **Given the following graph Gamma, give the shortest Hamiltonian cycle.**

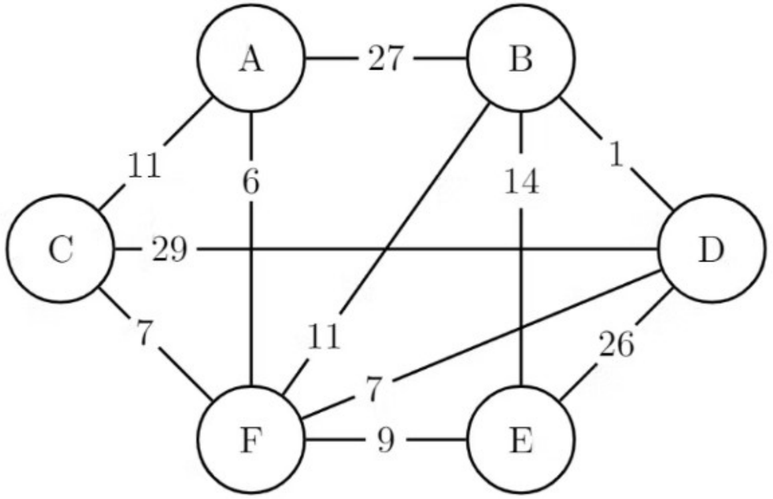
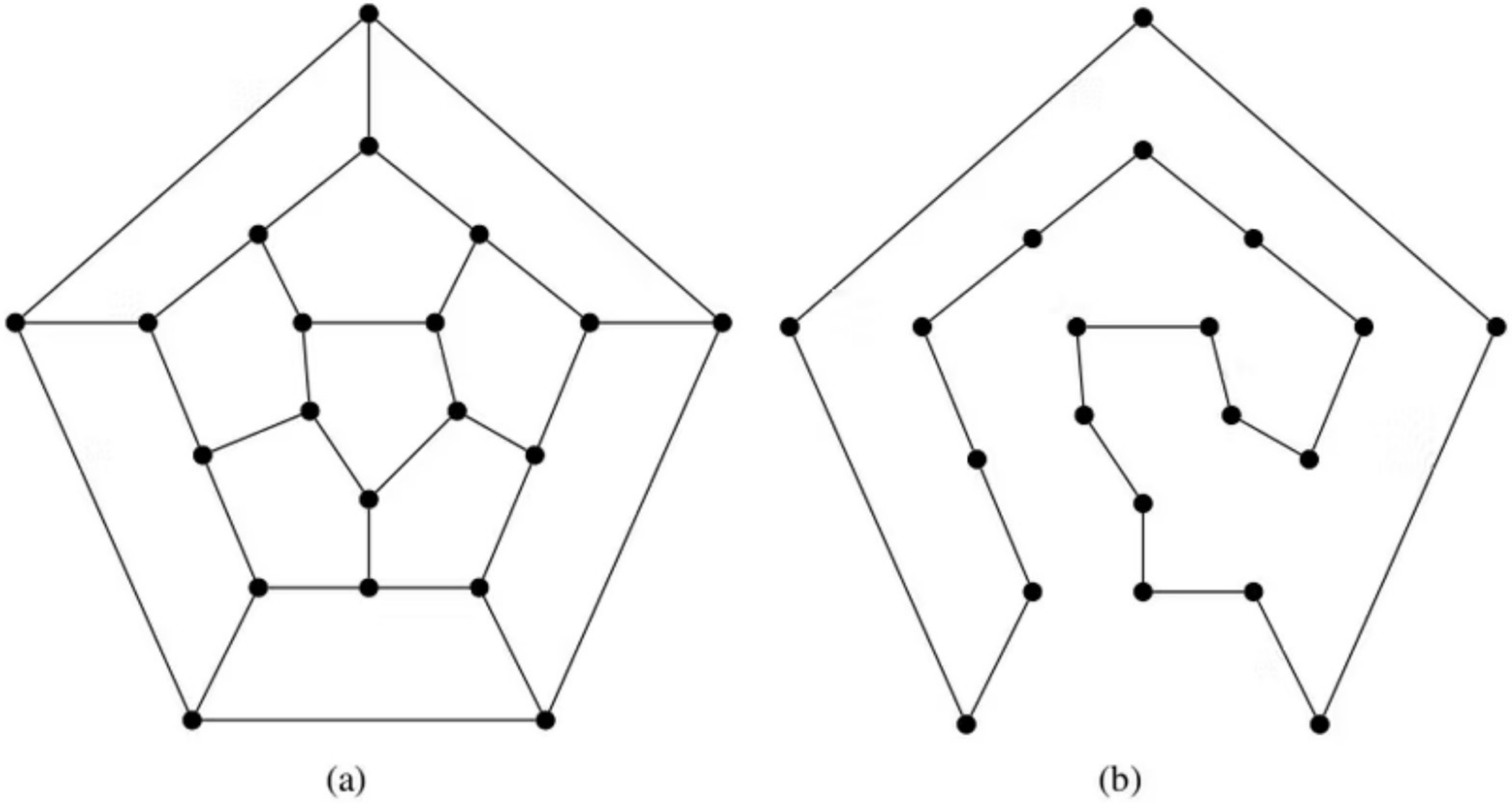


Figure 1: Gamma

Task 3

The Travelling Salesman Problem



A Hamilton Cycle (b) in the graph (a)

Task 4 Partitoning problems

a) Explain what a perfect three-dimensional matching is

b) Is graph coloring with $k = 2$ NP-complete? Explain your answer

Graph Coloring

- More specifically: vertex coloring
- Assign a color to each vertex
- No two adjacent vertices may have the same color
- k-coloring: use only k number of colors

Assignment 7 - NP and Computational Intractability

Part 2 - Coloring

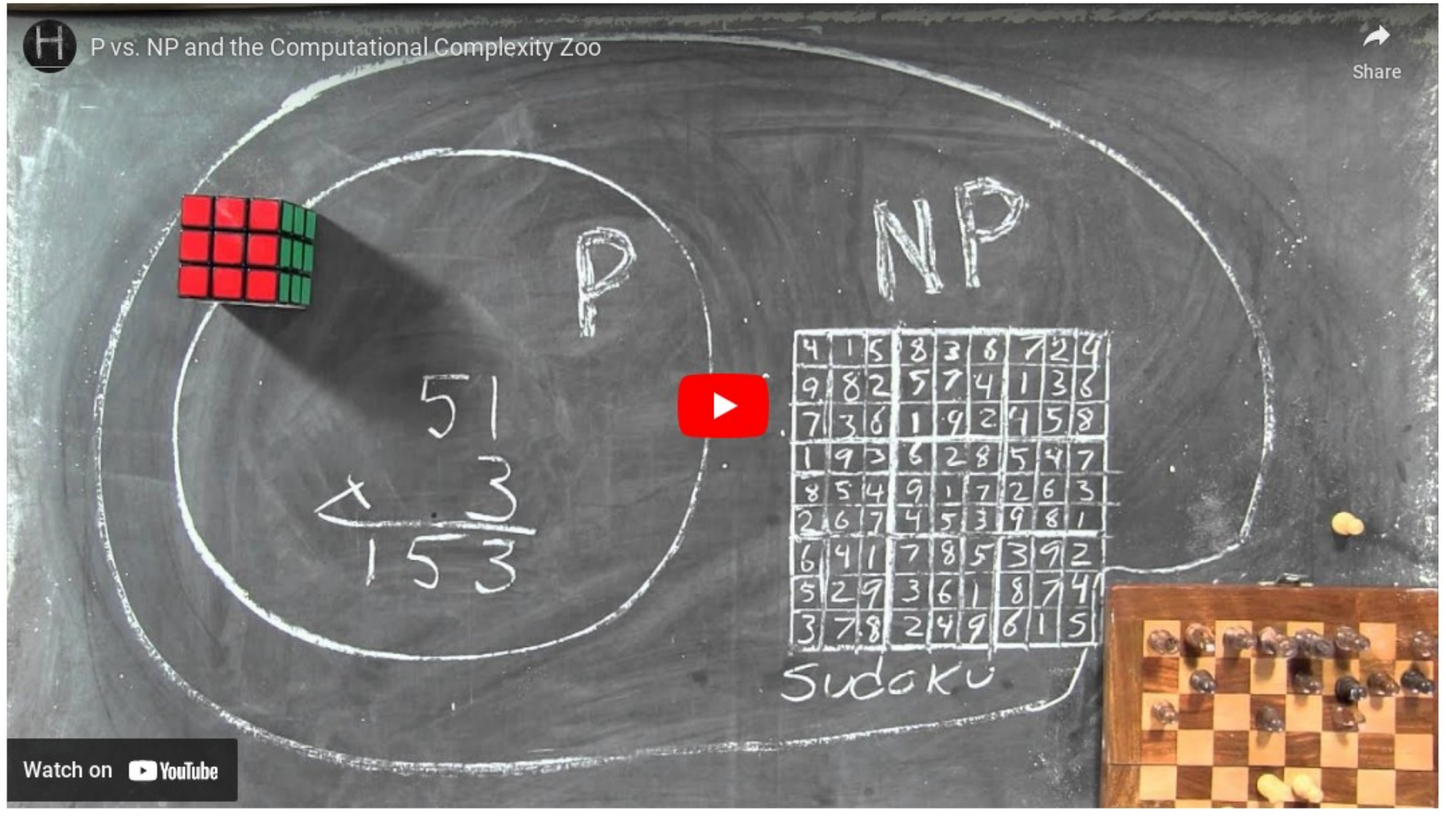
Problem Description

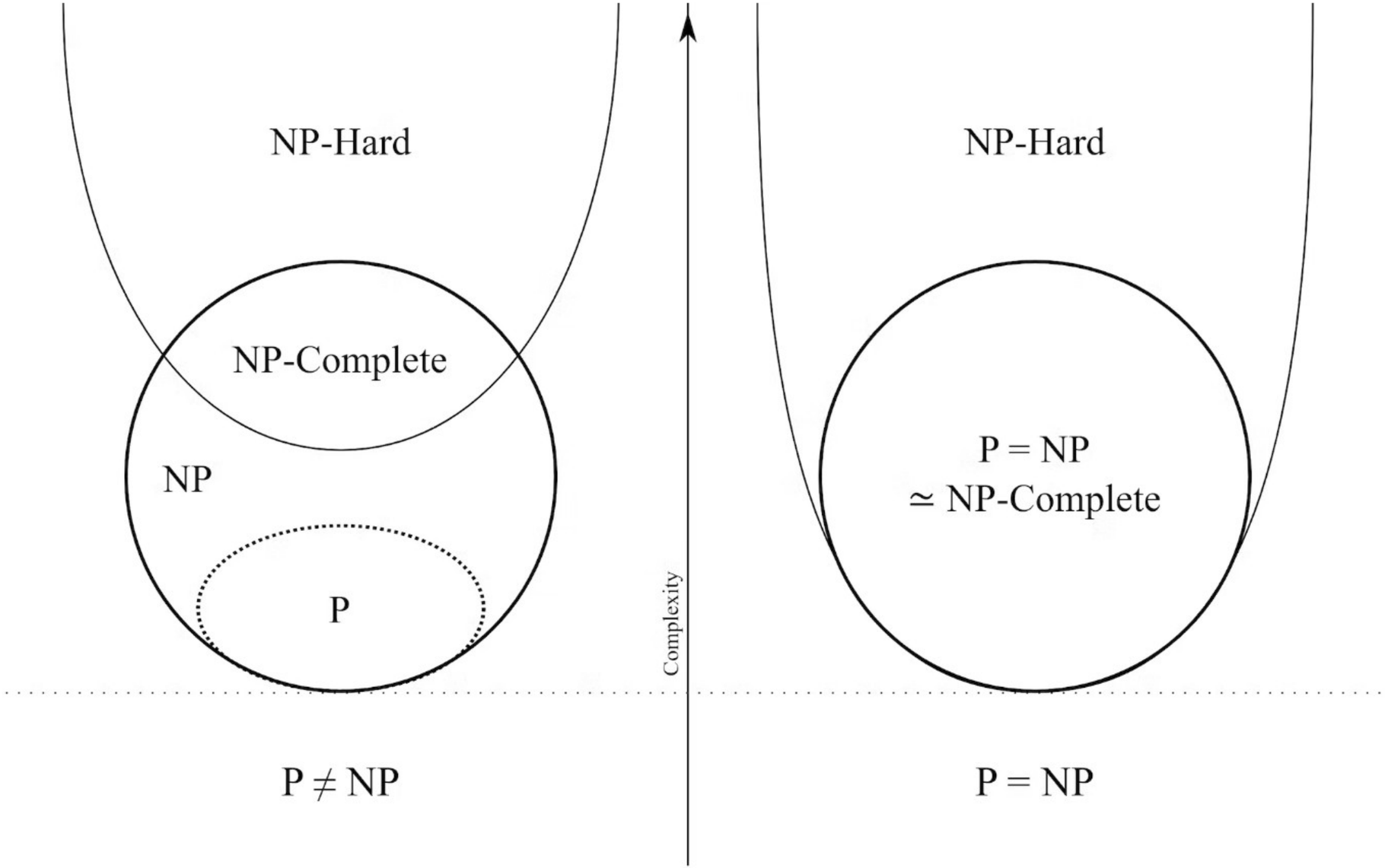
We want to create a naive recursive solution to the graph coloring problem. We will imagine the graph as a map of cities, and we want to color it in with k different colors without any two bordering cities having the same color. We will represent the map with an adjecency matrix.

This matrix will be a 2D python list of `1` and `0` (can be interpreted as `True` and `False`) where if `city 0` is a neighbor with only city `2`, its list of neighbors will be `[0, 0, 1, 0, ...]`

We'll also keep track of the colors using a basic list of integers, where different numbers represent different colors. When the algorithm finishes `city 0`, will use the color with index `0` from this list etc.

Part 2: Programming


P vs. NP and the Computational Complexity Zoo
Share
P
NP
51
3
153
Sudoku
Watch on YouTube


NP-Hard
NP-Complete
NP
P
Complexity
NP-Hard
P = NP
≃ NP-Complete
P ≠ NP
P = NP

What if P = NP?

> I know an NP-Complete joke, but once you've heard one you've heard them all.