Introduction to theory of computing via graph coloring

Tuukka Korhonen

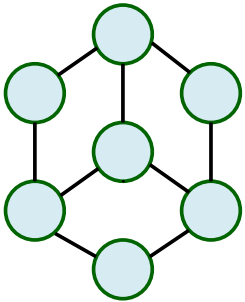
16 April 2026

IUP Lecture

Graphs and colorings

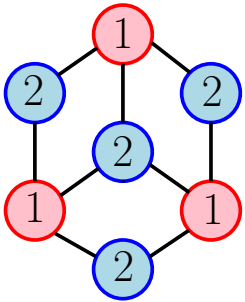
Graphs and colorings

- A graph consist of nodes (circles) connected by edges (line segments)



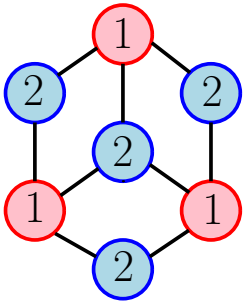
Graphs and colorings

- A graph consist of nodes (circles) connected by edges (line segments)
- Graph coloring: Give colors to nodes, so that for each edge, the nodes connected by it receive different colors



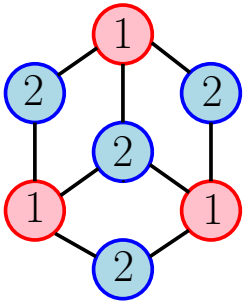
Graphs and colorings

- A graph consist of nodes (circles) connected by edges (line segments)
- Graph coloring: Give colors to nodes, so that for each edge, the nodes connected by it receive different colors
- 2-coloring problem: Is there a coloring with only 2 different colors?

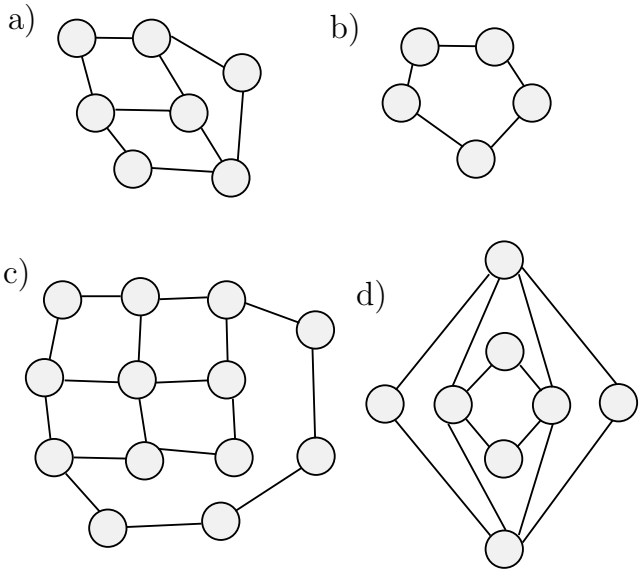


Graphs and colorings

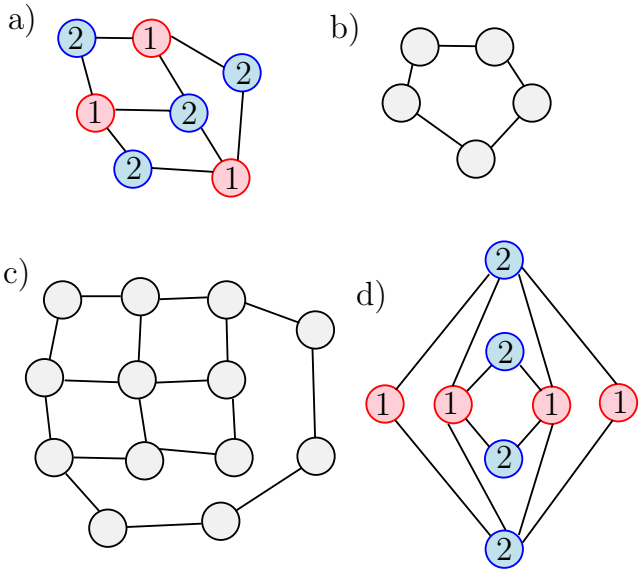
- A graph consist of nodes (circles) connected by edges (line segments)
- Graph coloring: Give colors to nodes, so that for each edge, the nodes connected by it receive different colors
- 2-coloring problem: Is there a coloring with only 2 different colors?
- Exercise 1: Consider the given graphs a-d, and find 2-colorings or that 2-coloring is not possible



2-coloring

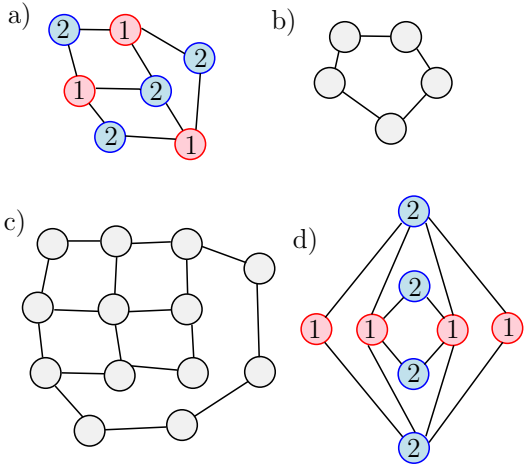


2-coloring



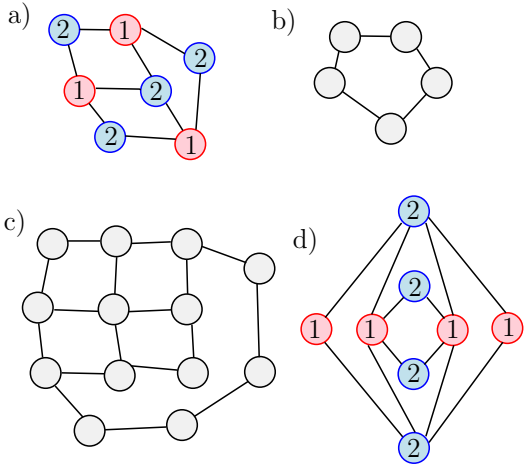
Algorithm for 2-coloring

- Did you come up with a method?



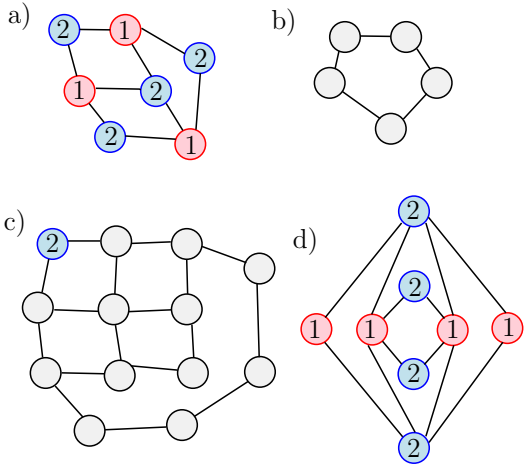
Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



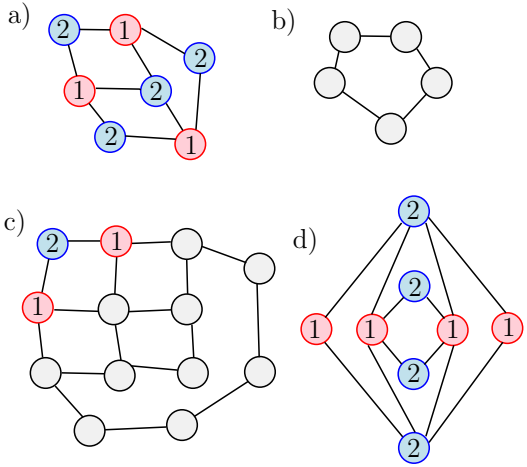
Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



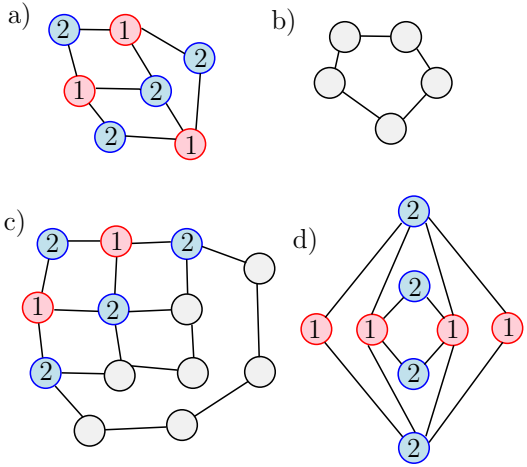
Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



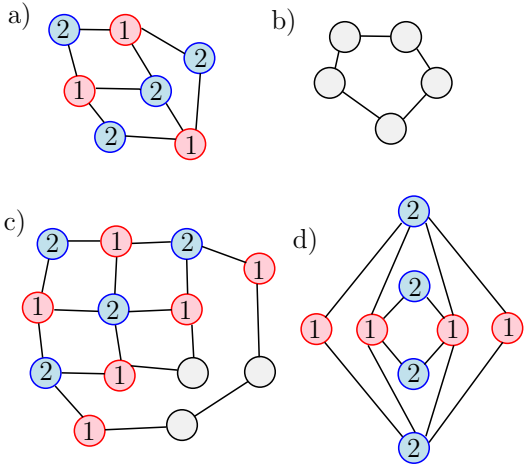
Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



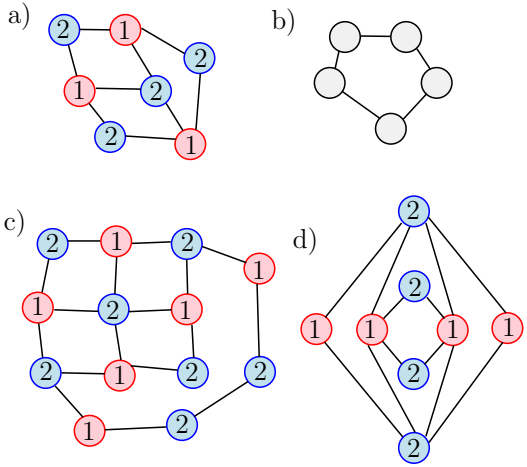
Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



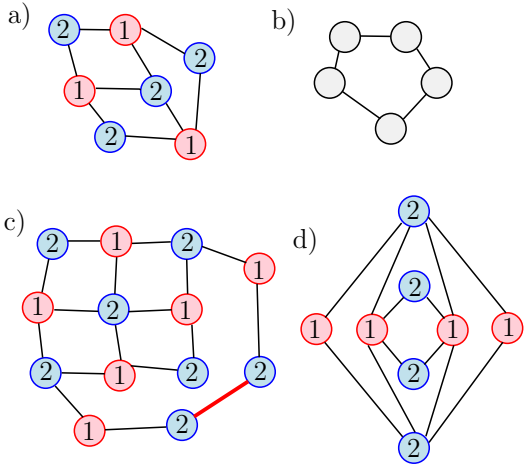
Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



Algorithm for 2-coloring

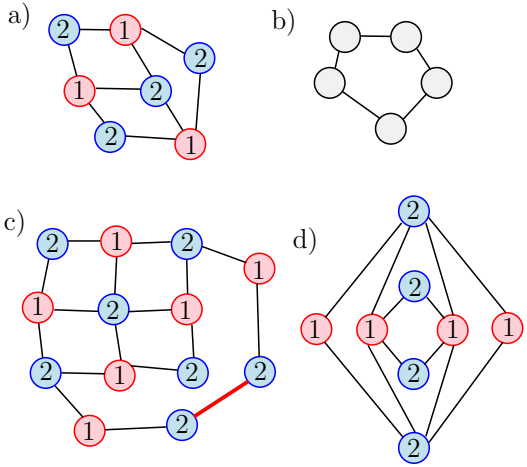
- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.



Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.

$\Rightarrow$ Efficient algorithm to test if a graph is 2-colorable

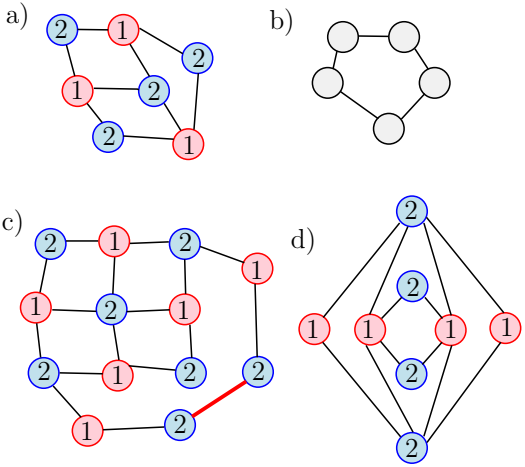


Algorithm for 2-coloring

- Did you come up with a method?
- Algorithm: Pick one node to be blue. Then infer the colors of others.

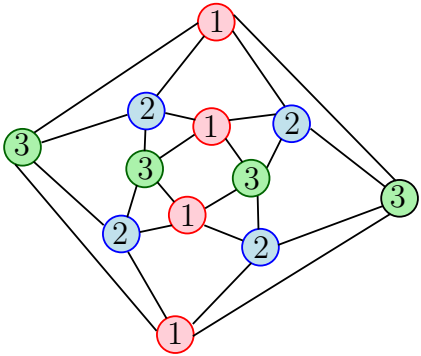
$\Rightarrow$ Efficient algorithm to test if a graph is 2-colorable

- Runs in $\approx n + m$ steps, where n the number of nodes, and m the number of edges



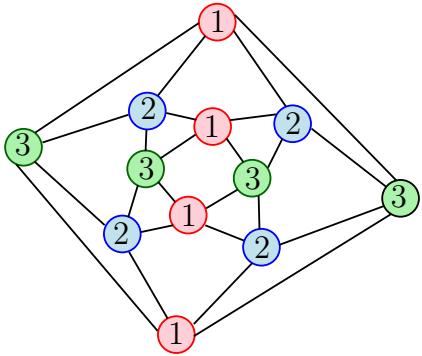
3-coloring

- How about 3-coloring?

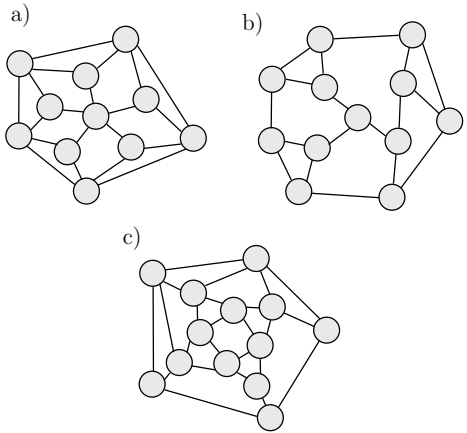


3-coloring

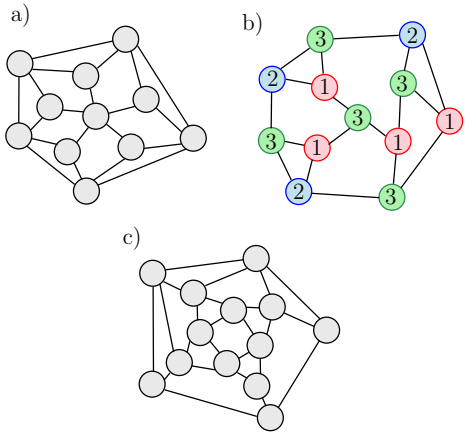
- How about 3-coloring?
- Exercise 2: Consider the given graphs a-c, and find 3-colorings or that 3-coloring is not possible



3-coloring

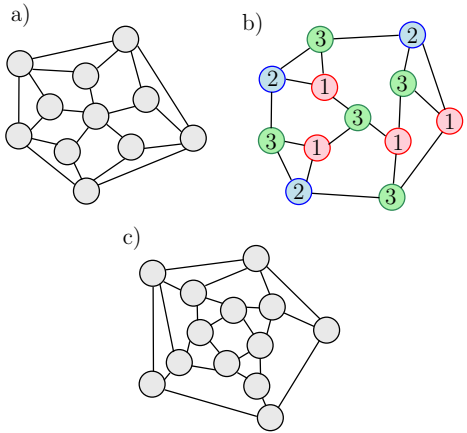


3-coloring



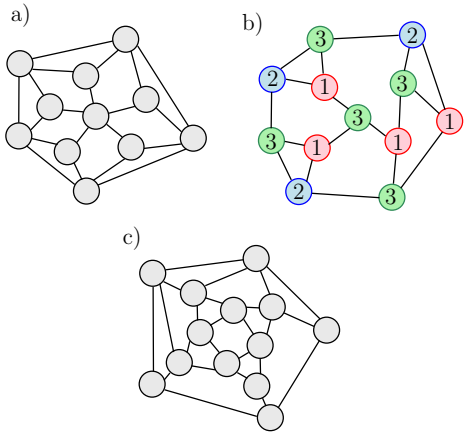
3-coloring

- Was 3-coloring more difficult?



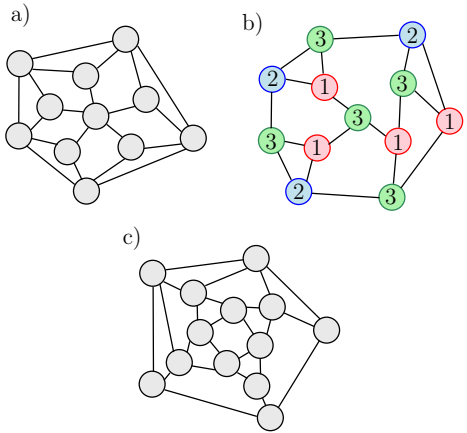
3-coloring

- Was 3-coloring more difficult?
- It should have been!



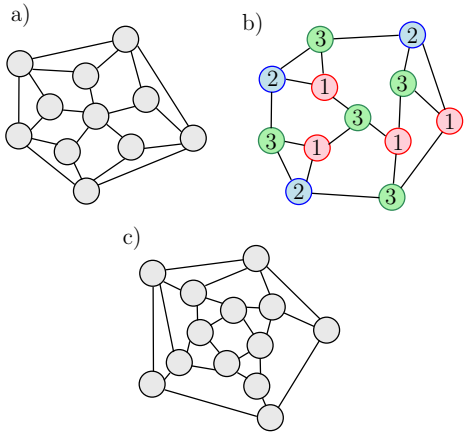
3-coloring

- Was 3-coloring more difficult?
- It should have been!
- According to theory of computing, there is no efficient algorithm for 3-coloring



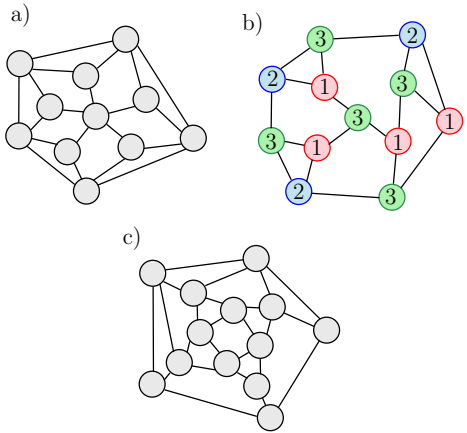
3-coloring

- Was 3-coloring more difficult?
- It should have been!
- According to theory of computing, there is no efficient algorithm for 3-coloring
- Any algorithm takes an exponential number of steps in the worst-case



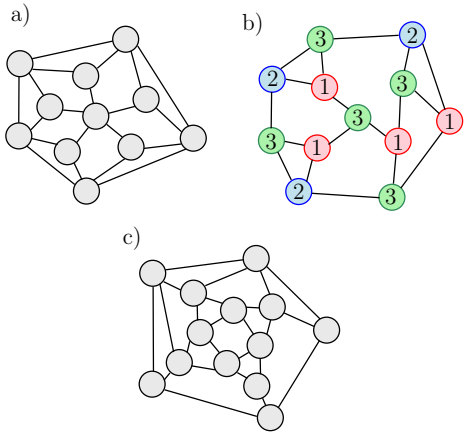
3-coloring

- Was 3-coloring more difficult?
- It should have been!
- According to theory of computing, there is no efficient algorithm for 3-coloring
- Any algorithm takes an exponential number of steps in the worst-case
- Trying all possibilities: $\approx 3^n$ steps in the worst-case



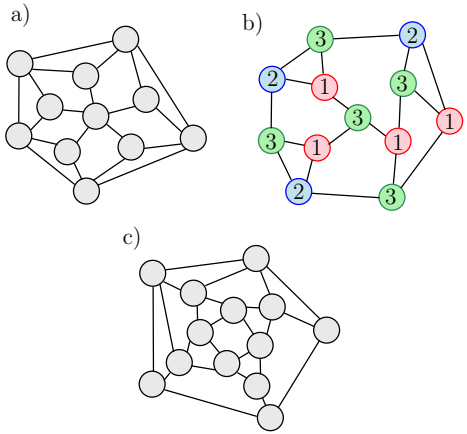
3-coloring

- Was 3-coloring more difficult?
- It should have been!
- According to theory of computing, there is no efficient algorithm for 3-coloring
- Any algorithm takes an exponential number of steps in the worst-case
- Trying all possibilities: $\approx 3^n$ steps in the worst-case
- Best known algorithm: $\approx 1.3289^n$ steps in the worst-case



3-coloring

- Was 3-coloring more difficult?
- It should have been!
- According to theory of computing, there is no efficient algorithm for 3-coloring
- Any algorithm takes an exponential number of steps in the worst-case
- Trying all possibilities: $\approx 3^n$ steps in the worst-case
- Best known algorithm: $\approx 1.3289^n$ steps in the worst-case
- Theory: There exists $\varepsilon > 0$, so that no algorithm solves 3-coloring with less than $(1 + \varepsilon)^n$ steps in the worst-case



Problems and algorithms

Problems that have efficient algorithms:

Problems where exponential time is required:

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring

Problems where exponential time is required:

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring

Problems where exponential time is required:

- 3-coloring

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities

Problems where exponential time is required:

- 3-coloring

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities
- Solving a system of linear inequalities

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities
- Solving a system of linear inequalities

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities
- Finding an integer solution to a system of linear inequalities

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities
- Solving a system of linear inequalities

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities
- Finding an integer solution to a system of linear inequalities
- Solving generalized Sudoku

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities
- Solving a system of linear inequalities
- Solving generalized Rubik's cube

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities
- Finding an integer solution to a system of linear inequalities
- Solving generalized Sudoku

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities
- Solving a system of linear inequalities
- Solving generalized Rubik's cube

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities
- Finding an integer solution to a system of linear inequalities
- Solving generalized Sudoku

– No easy way of telling which category a problem falls into

Problems and algorithms

Problems that have efficient algorithms:

- 2-coloring
- Finding the shortest route between two cities
- Solving a system of linear inequalities
- Solving generalized Rubik's cube

Problems where exponential time is required:

- 3-coloring
- Finding the shortest route that visits all cities
- Finding an integer solution to a system of linear inequalities
- Solving generalized Sudoku

– No easy way of telling which category a problem falls into

– Theory of computing: Finding efficient algorithms or proving that no efficient algorithms exist

Intended learning outcomes (ILOs)

- Design an efficient algorithm for 2-coloring a graph

Intended learning outcomes (ILOs)

- Design an efficient algorithm for 2-coloring a graph
- Explain that according to theory of computing, there is no efficient algorithm for 3-coloring

Intended learning outcomes (ILOs)

- Design an efficient algorithm for 2-coloring a graph
- Explain that according to theory of computing, there is no efficient algorithm for 3-coloring
- Explain that computational problems can be classified into categories based on whether efficient algorithms exist