DM898 Parameterized Algorithms
Fall 2025

Exercise Sheet 1

Complete before class on Wednesday, September 10th

Recap of complexity theory

Exercise 1 (Solved together during class)**.** Describe the following terms and give examples for them:

a. Optimization problem, decision problem

b. YES-instance, NO-instance

c. P, NP

d. NP-hard

e. NP-complete

f. Polynomial time reduction

Exercise 2 (Solved together during class)**.** In (decision version of) the *Clique problem*, we are given a graph $G = (V, E)$ and a number k. Our goal is to determine if there is a set $C \subseteq V$ with $|C| \geq k$ and $(u, v) \in E$ for each $u, v \in C$.
In (decision version of) the *Independent Set problem*, we are given a graph $G = (V, E)$ and a number k. Our goal is to determine if there is a set $I \subseteq V$ with $|I| \geq k$ and $(u, v) \notin E$ for each $u, v \in C$.
The Clique problem is NP-complete. Show that:

a. the Independent Set problem is NP-complete

b. the Vertex Cover problem is NP-complete

Branching and preprocessing

Exercise 3. Sudoku is a popular puzzle where you have to fill out the empty grid cells so that each row, column and 3x3 box contains all numbers $1, 2, \ldots, 9$.

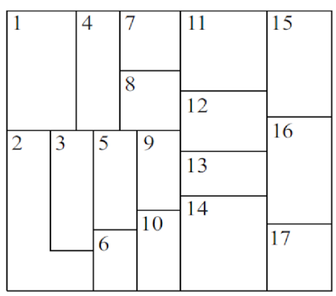
	3							
			1	9	5			
		8					6	
8				6				
4			8					1
				2				
	6					2	8	
			4	1	9			5
							7	

Source: https://commons.wikimedia.org/wiki/File:Sudoku_problem_1.svg

Consider an instance encoded as follows: for each grid cell you are given the subset of the numbers $\{1, 2, \ldots, 9\}$, which are the options left. In the beginning each empty cell has the complete set $\{1, 2, \ldots, 9\}$ and a cell filled i has the singleton set $\{i\}$. When you solve a Sudoku you probably implicitly follow a strategy of applying preprocessing rules to exclude certain options and branching in case you are stuck.

a. Describe precisely an algorithm solving Sudokus based on preprocessing rules and branching. Explain your design decisions, e.g. what to branch on.

Exercise 4. The Four Color Theorem states that the regions of any map (like below) can be colored with at most 4 colors in such a way that no two regions that share a border will be colored the same.



a. Model the problem of coloring a map with the least colors as a vertex coloring problem and write down the graph corresponding to the map above.

b. Using branching (and possibly preprocessing), determine whether the map above can be colored with 3 colors.

Hint: for (b), think about the following questions. What can you say about a vertex with degree at most 3? Instead of branching on the color of a vertex, you might also branch on whether two vertices have the same color. How would the corresponding subproblems look like? When can you be sure that an instance requires at least 4 colors?

Dynamic programming

Exercise 5. An algorithms that solves vertex coloring by complete enumeration would require $c^n \cdot n^{O(1)} \leq n^n \cdot n^{O(1)}$ time. Here, c is the optimal number of colors. Using dynamic programming over subsets, give an algorithm with running time $3^n \cdot n^{O(1)}$.