

Assignment 5

Responsible TA: Claudi Lleyda Moltó

Delivery instructions

To deliver an assignment you have to make sure you deliver both parts. You can deliver as many times as you want before the deadline on Blackboard. Delivery should be done in two separate files according to the following specifications.

The theoretical question answers must be uploaded as PDF, how you answer the questions (handwritten scans, Word, L^AT_EX, etc.) doesn't matter, as long as they are delivered as a single PDF file and the answers are readable. The filename should contain your NTNU username and the assignment number, for instance: `karinor-05.pdf`.

The notebook should be delivered in full (as `.ipynb`). That way the TAs can easily run your code and verify the output. The filename should contain your NTNU username and the assignment number, for instance: `karinor-05.ipynb`.

Theory

Task 1 - Interval cover

Given a set x_1, x_2, \dots, x_n of points on the real line, we want to find the smallest set of unit-length closed intervals that contains all of the given points.

Find a greedy algorithm that solves this problem optimally. Justify your answer.

Task 2 - Unique minimum spanning tree

Let G be a weighted graph in which all the edges have distinct weights. Prove or disprove that G has a unique minimum spanning tree.

Task 3 - Interval graph coloring

Given a set of intervals, we define a graph whose vertices are intervals, and whose edges connect intersecting intervals. A coloring of a graph is an assignments of colors to the vertices such that no two adjacent vertices have the same color. Describe a greedy algorithm to color an interval graph with the fewest amount of colours possible. Justify your answer.

Programming

This part should be solved in the corresponding Jupyter Notebook. Refer to the notebook for further details and instructions regarding the programming tasks.

Assignment submission

The assignments will be given out on Blackboard where each assignment has a written theoretical part and a programming part in Python using Jupyter Notebook. The notebook task can be solved locally (for instance in VSCode), or you can upload the notebook to JupyterHub and run the code there.

The notebooks will guide you through a, more or less, specific solution to a problem. You are however encouraged to attempt to implement your own/other solutions, as long as

- you don't use any additional libraries or packages,
- you don't alter the input values or structures, and
- your solution solves the problem described in the assignment.