Assignment 3

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.

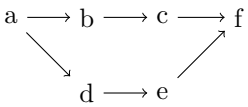
The theoretical question answers must be uploaded as PDF, how you answer the questions (handwritten scans, Word, LaTeX etc.) doesn't matter, as long as they are delivered as a single PDF file and the answers are readable. The filename should contain your NTNU username and the assignment number, for instance: `karinor-03.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-03.ipynb`.

Theory

Task 1 - Topological ordering

How many topological orderings does the following graph have?



Task 2 - Binary trees

A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree the number of nodes with two children is exactly one less than the number of leaves.

Task 3 - Matching DFS and BFS

Let $G = (V, E)$ be a connected graph and $u \in V$ one of its vertices. Suppose we compute a depth-first search tree rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute a breadth-first search tree rooted, again, at u, and obtain the same tree T. Prove that $G = T$.

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 uploaded the notebook to JupyterHub and run the code there.

The notebooks will guide you though 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.