ADS 1 — problem sheet 3

Excercise 1: How to determine if given graph has precisely one topological order?

Excercise 2: Design $O(n + m)$ algorithm for topological sorting that is based on the idea of repeated removal of sources.

Excercise 3: Describe algorithm to find a shortest path in a edge-labelled DAG which makes use of the topological ordering of vertices. *Edge-labelled DAG* an acyclic oriented graph $G = (V, E)$ where every edge $e \in E$ has associated *length* $\ell(e)$. The *length* of a given path is then sum of lengths of all edges in it. On next class we will study an algorithm to answer this question on geraphs in general.

Excercise 4: Do the same to find the longest path in an edge-labelled DAG (this is a problem we do now know how to solve effectively on general graphs).

Excercise 5: We say that connected graph $G = (V, E)$ is *semi-connected* if for every pair of vertices $u, v \in E$ there exists an oriented path from u to v or from v to u (possibly both). Design linear-time algorithm to decide if given graph G is semi-connected..

Homework 2: Construction of a house consists of multiple tasks where some needs to be finished before others can start. We describe this as a graph: vertices are tasks and each vertex has associated duration. Oriented edges represents dependencies. Write algorithm to determine shortest time in which the house can be constructed (assuming there is unlimited number of workers doing individual tasks) and find critical ones (that is, tasks whose delays would result in longer overall construction times).