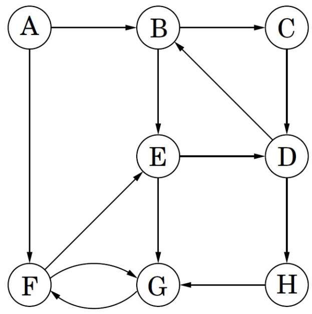
3. Werkcollege 3

You are allowed to answer in Dutch.

3.1. Perform depth-first search on the following graph; whenever there's a choice of vertices, pick the one that is alphabetically first. Classify each edge as a tree edge, forward edge, back edge, or cross edge.



3.2. Suppose a CS curriculum consists of n courses, all of them mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w. Find an algorithm that works directly with this graph representation, and computes the minimum number of semesters necessary to complete the curriculum (assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear.

3.3. You are given a rooted binary tree $T = (V, E)$, with a designated root node $r \in V$. We say that u is an ancestor of v in the rooted tree if the unique path from r to v in T passes through u. You wish to preprocess the tree so that queries of the form is u an ancestor of v? can be answered in constant time. The preprocessing itself should take linear time. How can this be done?

3.4. Most graph algorithms that take an adjacency-matrix representation as input require time $\mathcal{O}(|V|^2)$, but there are some exceptions. Show how to determine whether a directed graph $G = (V, E)$ contains a *universal sink*, i.e., a vertex with in-degree $n1$ and out-degree 0, in time $\mathcal{O}(|V|)$ given an adjacency matrix for G.

3.5. Give an efficient algorithm which takes as input a directed graph $G = (V, E)$, and determines whether or not there is a vertex $s \in V$ from which all other vertices are reachable.

3.6. Another way to perform topological sorting on a directed acyclic graph $G = (V, E)$ is to repeatedly find a vertex of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Explain how to implement this idea so that it runs in time $\mathcal{O}(|V| + |E|)$.