TD 4: Connectivity Applications

1 Longest Path on a DAG

Give an algorithm which takes as input a DAG G and outputs the longest directed path in G. Consider using a greedy algorithm. Is greediness a good idea?

2 Count Paths on DAGs

Design an algorithm that takes as input a DAG G and two vertices s, t and outputs the number of distinct paths from s to t in G. At most how many such paths may exist in a DAG with n vertices? **NB:** Your algorithm does not need to list the paths, just output their number.

3 Alternative Topological Sort

In class we saw a linear-time algorithm for topologically sorting a DAG based on DFS. For this exercise we will design an algorithm from first principles. Show that a DAG can be sorted in linear time by computing the indegree of each vertex and then repeatedly removing sources and updating the indegrees as needed.

4 A bridge too far

In class we saw that an articulation point is a vertex whose removal disconnects the graph (more precisely, increases the number of components). The equivalent notion for edges is that of a **bridge**: an edge e is a bridge if removing it from a graph increases the number of connected components.

Give an algorithm which takes as input a connected undirected graph and identifies all bridges.