

Let $G = (V, E)$ be a directed and acyclic graph. We will refer to these as DAGs, and they will become important structures later on. In this problem, we will prove some basic results about DAGs.

- (a) Show that there exist a vertex in G that has no incoming edges. We refer to these as *source* vertices.
- (b) Show that there exist a vertex in G that has no outgoing edges. We refer to these as *sink* vertices.
- (c) Show that there exist a mapping $f : V \rightarrow \{1, \dots, n\}$ such that $(u, v) \in E$ implies that $f(u) \leq f(v)$.

Another way to interpret this is that this mapping *respects* the direction of the edges. We refer to this ordering as a *topological order* of G . You will later see that we can *sort* these vertices by its topological order; this is called a *topological sort*.

Solution.

- (a) Suppose that every vertex had an incoming edge. Then every possible path will start at a vertex with an incoming edge. We can extend this path indefinitely, by repeatedly prepending this incoming edge to the path. Therefore at some point, the number of vertices in the path exceeds the number of vertices in the graph. Using the pigeonhole principle, some vertex must have appeared in the path at least twice, indicating the presence of a cycle, which is a contradiction.
- (b) Suppose that every vertex had an outgoing edge. Then every possible path will end on a vertex with an outgoing edge. We can extend this path indefinitely, by repeatedly appending this outgoing edge to the path. Therefore at some point, the number of vertices in the path exceeds the number of vertices in the graph. Using the pigeonhole principle, some vertex must have appeared in the path at least twice, indicating the presence of a cycle, which is a contradiction.
- (c) We can do a proof by construction by describing and justifying an algorithm used to construct a topological order, namely Kahn's algorithm. While there are still vertices in the graph, pick any vertex with no incoming edges, assign the mapping of the vertex to the index, increment the index, and then remove it from the DAG.

The resultant mapping f satisfies the requirements. Suppose that there existed some $(u, v) \in E$ such that $f(u) > f(v)$. This means that v was removed from the graph before u . However this is a contradiction, because when v is removed, there was at least one incoming edge (u, v) .