1. Inspired by the previous lab, you decide to organize a Snakes and Ladders competition with n participants. In this competition, each game of Snakes and Ladders involves three players. After the game is finished, they are ranked first, second, and third. Each player may be involved in any (non-negative) number of games, and the number need not be equal among players.

   At the end of the competition, m games have been played. You realize that you forgot to implement a proper rating system, and therefore decide to produce the overall ranking of all n players as you see fit. However, to avoid being too suspicious, if player A ranked better than player B in any game, then A must rank better than B in the overall ranking.

   You are given the list of players and their ranking in each of the m games. Describe and analyze an algorithm that produces an overall ranking of the n players that is consistent with the individual game rankings, or correctly reports that no such ranking exists.

   Solution: We reduce to the topological sorting problem in a directed acyclic graph $G = (V, E)$ as follows:

   - V is the set n of players.
   - E contains a directed edge $i \to j$ if player i ranked higher than player j in any game. Since each game ranks three pairs of players, E contains $3m$ edges.
   - No additional values are associated with the nodes or edges.
   - We need to compute a topological order for this dag or report correctly that no such order exists.
   - We can find a topological order using depth-first search.
   - The algorithm runs in $O(V + E) = \boldsymbol{O(n + m)}$ ***time***.

   ■

2. Given a directed graph $G = (V, E)$, two distinct nodes s, t are said to incomparable if niether s can reach t nor t can reach s.

   - Draw a DAG on 4 nodes where there is *no* incomparable pair. Draw a DAG on 4 nodes where there is an incomparable pair and the DAG has only one source and only one sink.
   - Describe a linear time algorithm to check whether a given DAG G has an incomparable pair of nodes.
   - Describe a linear time time algorithm for the same problem in a general directed graph.

Solution: For the first two parts consider vertex set $V = \{a, b, c, d\}$. A DAG with no incomparable pairs is a path $a \to b \to c \to d$. Now consider the DAG on same vertex set with edges $\{(a, b), (a, c), (b, d), (c, d)\}$. b, c form an incomparable pair and a is the unique source and d is the unique sink.

Compute a topological sort of G in linear time using DFS or the algorithm that peels off one source at a time. Suppose the ordering is $v_1, v_2, \ldots, v_n$. If there is any $1 \leq i < n$ such that (v_i, v_{i+1}) is *not* an edge in G then we see that v_i and v_{i+1} is an incomparable pair. We leave the formal proof as an exercise. One can check whether there is any such i in linear time (how?). If there is no such i then there is no incomparable pair. To see this, consider v_i and v_j where $i < j$. We see that v_i can reach v_j via the path $v_i \to v_{i+1} \to \ldots v_j$.

For a general directed graph G we first compute the meta-graph graph G^{SCC} of G in linear time using the algorithm described in lecture. We observe that G has an incomparable pair if and only if G^{SCC} has an incomparable pair. Assuming this claim we can run the algorithm for DAGs on G^{SCC}.

We elaborate on the claim. Let $S_1, S_2, \ldots, S_\ell$ be the strong-connected components of G and let $v_1, \ldots, v_\ell$ be the vertices in G^{SCC} that represent these components. If v_i and v_j are incomparable in G^{SCC} then we claim that for any vertex $a \in S_i$ and vertex $b \in S_j$ the pair a, b is incomparable in G. Similarly, if $a, b \in V(G)$ are incomparable in G and $a \in S_i$ and $b \in S_j$ then v_i and v_j are incomparable in G^{SCC}. We leave the formal proofs of these sub claims as exercises for you. ■