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.

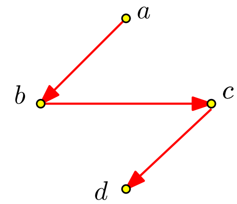
Solution: We reduce to the topological sorting problem in a directed acyclic graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ as follows:

- V is the set n of players.
- E contains a directed edge $i \rightarrow j$ 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 compute a topological order in linear time.
- The algorithm runs in $O(|V| + |E|) = O(n + m)$ time.

2 Given a directed graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, two distinct nodes s, t are ***incomparable*** if neither s can reach t nor t can reach s.

2.A. Draw a DAG on 4 nodes where there is *no* incomparable pair.

Solution: Consider vertex set $V = \{a, b, c, d\}$. A DAG with no incomparable pairs is a path $a \to b \to c \to d$.

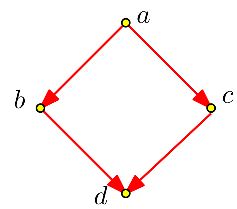


2.B. Draw a DAG on 4 nodes where there is an incomparable pair and the DAG has only one source and only one sink.

Solution: Consider the vertex set $V = \{a, b, c, d\}$, with the set of edges being

$$\{(a,b), (a,c), (b,d), (c,d)\}.$$

The vertices b, c form an incomparable pair and a is the unique source and d is the unique sink.



2.C. Describe a linear time algorithm to check whether a given DAG G has an incomparable pair of nodes, and if so output it.

Solution:

The algorithm. 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 $v_1, v_2, \ldots, v_n$. The algorithm scans, in $O(n+m)$ time, all the edges, and verifies that the edge $v_i \to v_{i+1}$ is in G, for all i. To this end, the algorithm initialize a binary vector $B[1 \ldots n-1]$ to FALSE. Now, when scanning all the edges, when encountering an edge $v_j \to v_{j+1}$, set $B[v_j]$ to TRUE. This takes $O(n+m)$ time. Now, scan the array B to verify all entries are TRUE. If they are, then all vertices are comparable. Otherwise, output the pair $v_j \to v_{j+1}$ for which $B[j]$ is false, as the incomparable pair.

Correctness.

Lemma 0.1. *If $v_i \to v_{i+1}$ is not an edge in G, for some i, then v_i and v_{i+q} are incomparable in G.*

Proof: Proof by contradiction. Let $s = v_i$ and $t = v_{i+1}$.

Recall, that all the edges in the graph G are of the form $v_x \to v_y$, for $x < y$. For an edge $e = (v_x, v_y) \in \mathsf{E}(\mathsf{G})$, let $\Delta(e) = y - x$. Observe that $\Delta(e) > 0$ for all edges in G.

Assume that s and t are comparable, and $s \prec t$. But then there is a path π of the form $s = u_1 \to u_2 \to \cdots \to u_k = t$ in G, for some $k \geq 2$. If $\Delta(u_1 \to u_2) = 1$ then there is an edge between v_i and v_{i+1}, which is impossible. As such, $\Delta(u_1 \to u_2) \geq 2$ and $k > 2$. We have that

$$1 = (i+1) - i = \Delta(\pi) = \sum_{t=1}^{k-1} \Delta(u_t \to u_{t+1}).$$

But as $k \geq 3$, and the first term in this summation is at least 2, this implies that one term must be negative. A contradiction.

As for the other case, if s and t are comparable, and $t \prec s$, then the same argument applies. There is a path σ between the two vertices, with $\Delta(\sigma) = -1$. Namely, the sum of the Δs on the edges of σ -1. As such, there must be an edge $e \in \sigma$ with $\Delta(e) < 0$, which is impossible. ■

The above lemma readily implies the correctness of the algorithm. If B is all true then G contains a Hamiltonian path and all vertices in the graph are comparable. Otherwise, there exists two consecutive vertices in the topological ordering that are not connected by edge, and the lemma implies that they are incomparable.

Algorithm II. Remove the vertices from the graph according to their incoming degrees (removing vertices with incoming degree 0). This can be done in $O(n + m)$ time as seen in class (and is another way to do topological sorting). If at any point in time there are two vertices with incoming degree 0, then this pair is incomparable, and algorithm stops and output it.

Algorithm III.

If the path length is $n - 1$, there is no comparable pair. However, this does not quite solve the problem, since it asks you to output the incomparable pair – one can still do it with a bit of thinking, but at these point the other solutions are simpler.

2.D. Describe a linear time algorithm for the same problem in a general directed graph.

<u>Solution</u>:

The algorithm. For a general directed graph G, we first compute the meta-graph $\mathsf{G}^{\mathrm{SCC}}$ of G in linear time using the algorithm described in lecture. As $\mathsf{G}^{\mathrm{SCC}}$ is a DAG, we topologically sort it, and let $S_1, \ldots S_k$ be the connected components of G numbered according to this topological ordering (they also form the vertices of $\mathsf{G}^{\mathrm{SCC}}$).

We run the algorithm from (2.C.) on G^{SCC} and return whatever answer it returns, as the desired answer.

Proof of correctness.

Lemma 0.2. *If there is no path from S_i to S_j in G^{SCC}, then there is no pair of vertices $s \in S_i$ and vertex $t \in S_j$ such that s can reach t in G.*

Proof: Assume for contradiction that s can reach t in G, and let π be this path. By shrinking edges in π that lies inside the same connected component of G, we end up with a path from S_i to S_j in $\mathsf{G}^{\mathrm{SCC}}$ which is impossible. ∎

If the algorithm of **(2.C.)** returns that there is an incomparable pair, then it also return the two components S_i and S_{i+1} that are incomparable. Pick any vertex $v_i \in S_i$ and $v_{i+1} \in S_{i+1}$, and observe that by the above lemma v_i can not reach v_{i+1}.

If the algorithm of **(2.C.)** returns that there is no incomparable pair, then the edges $S_i \to S_{i+1}$ are all in $\mathsf{G}^{\mathrm{SCC}}$, for all i. For each edge $S_i \to S_{i+1}$, pick the two vertices $v'_i \in S_i$ and $v_{i+1} \in S_{i+1}$, such that $v'_i \to v_{i+1} \in \mathsf{E}(\mathsf{G})$. Since each v_i can reach v'_i, and vice versa, since they are both in the same connected component. It follows that any vertex v_i can reach any vertex v_j, if $i < j$.

This readily implies that all pairs of vertices in the graph are "comparable". Indeed, consider two vertices $x, y \in \mathsf{V}(\mathsf{G})$, and assume that $x \in S_i$ and $y \in S_j$, for $i < j$. There is a path from x to v_i, and v_j to y, since both pairs are in the same strong connected component. Since there is a path from v_i to v_j in G, it readily follows that there is a path from x to y in G, as claim.