For each of the problems below, transform the input into a graph and apply a standard graph algorithm that you have seen in class. Whenever you use a standard graph algorithm, you ***must*** provide the following information. (I recommend actually using a bulleted list.)

- What are the vertices?
- What are the edges? Are they directed or undirected?
- If the vertices and/or edges have associated values, what are they?
- What problem do you need to solve on this graph?
- What standard algorithm are you using to solve that problem?
- What is the running time of your entire algorithm, *including* the time to build the graph, *as a function of the original input parameters*?

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.

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.

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.

2.C. Describe a linear time algorithm to check whether a given DAG G has an incomparable pair of nodes.

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