Exercises - Graph Introduction
Tutorial

1. **Prove the White-Path Theorem:** Recall the White-Path Theorem: *In a depth-first-forest of a Graph G, vertex v is a descendant of vertex u if and only if at the time that is discovered by DFS, vertex v can be reached from u along a path consisting entirely of white vertices.* Prove that the White-Path Theorem is correct.

2. **Prove the correctness of topological sort:** Recall the topological sort algorithm introduced in the lecture that runs a DFS and returns the vertices in descending order of their finished time. Show that the algorithm correctly sorts the vertices in a topological order.

3. **Semi-connected graph:** A directed graph $G = (V, E)$ is semi-connected iff for each pair u and v of V, there is a path from u to v or there is a path from v to u.

   (a) Assume G has two vertices u and v, both of which have in-degree 0. Prove that G is not semi-connected.

   (b) Assume G has two vertices u and v, both of which have in-degree 1 and assume (w, u) and (w, v) are two arrows in G where $w \in V$ has in-degree 0 using 3a. Prove that G is not semi-connected.

   (c) Assume G is a DAG. Design an $O(|V|^2)$-time algorithm to decide whether G is semi-connected or not using 3a and 3b.

   (d) For arbitrary directed graph G, design an $O(|V|^2)$-time algorithm to decide whether G is semi-connected or not (use 3c).

   (e) Let $\mathcal{C}$ be the set of the strongly connected components of G. Let $G^* = (V^*, E^*)$ be the directed graph such that there is a corresponding vertex $v_C \in V^*$ for each component $C \in \mathcal{C}$, and $(v_C, v_{C'}) \in E^*$ if there is a path from a vertex in C to a vertex in C'. Obtain the sorted vertices $(v_1, v_2, \ldots)$ by running topological sort on G^*. Prove that $(v_i, v_{i+1}) \in E^*$ for $1 \leq i < |V^*|$ iff G is semi-connected. Use this property to design a faster algorithm that decides if G is semi-connected.

4. **Rolling Die Game:** Consider rolling a die on a grid plane, where each entry on the plane may be labeled "forbidden" at the beginning. Meanwhile, the bottom-left and top-right entries are not forbidden, but with labels s and t, respectively, where $s, t \in \{1, 2, 3, 4, 5, 6\}$. The die is initially put at the bottom-left entry with the side s up. At any round, the die can be rolled up or rolled right to any neighboring entries that are not labeled as forbidden. The goal is to decide if it is possible to move the die by rolling it up or right such that it is eventually placed at the top-right corner with the side t up.

   Model the rolling die game as a graph problem and describe how to solve it. Note that you can make the decision on how the die was placed in the first place, as long as it has the side s up.