Assignment 7

Module: Graphs

Instructions:

- Honor code: Work on this assignment with at most one partner. Between different teams, Collaboration is at level 1 [verbal collaboration only]
- Write each problem on a separate page; If a problem has multiple parts, you can write all parts on the same page, as long as you leave space in between them.

---

1. Assume you are given a DAG (directed acyclic graph) G, and you want to compute *longest* paths rather than shortest. The edges do not have weights; the length of a path is the number of edges on the path.

   (a) Given a vertex u in G, describe how to compute the longest path from u. Ideally your algorithm will run in $O(V + E)$ time (Hint: dynamic programming).

   (b) Describe how to compute the longest path in G. Ideally your algorithm will run in $O(V + E)$ time.

   Note: The problem of determining the *longest path* is known to be NP-complete on arbitrary graphs. On DAGs it can be solved in linear time.

2. Given a DAG, design a linear time algorithm to determine whether there is a directed path that visits each vertex exactly one.

   Notes: In an undirected graph G: A path that visits each vertex exactly once is called a *Hamiltonian path.* A cycle that visits each vertex once is called a *Hamiltonian cycle*, and a graph that has a Hamiltonian cycle is called a *Hamiltonian graph.* The problem of determining whether an arbitrary graph has a Hamiltonian path/cycle is known to be NP-complete. On DAGs this problems can be solved in linear time.