

Assignment 10

Algorithms, Spring 2024

Honor code: *Work on this assignment alone or with one partner (highly encouraged). Partner policy: You and your partner will work together on the assignment throughout the whole process, you will write it and review it together, and will submit one assignment. You can talk to anyone in the class (collaboration level 1). It is not allowed to search online for the specific problems in this assignment—doing so violates academic honesty for the class.*

1. **Longest paths in DAGs:** You are given a DAG (directed acyclic graph) G . In this problem you want to compute *longest* paths rather than shortest. The edges in G do not have weights and the length of a path is defined as the number of edges on the path.
 - (a) Given a vertex u in G , describe how to compute the longest path from u (to any vertex in G). Ideally your algorithm will run in $O(|V| + |E|)$ time.
 - (b) Describe how to compute the longest path in G . Ideally your algorithm will run in $O(|V| + |E|)$ time.

We expect: pseudocode, justification and run time analysis.

Notes: the problem of determining the *longest path* is known to be NP-complete on arbitrary graphs. But, on DAGs, longest paths can be found in linear time, which you'll do in this problem!

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

We expect: pseudocode, justification, analysis.

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. But, on DAGs this problems can be solved in linear time, which you'll do in this problem!