CS3000: Algorithms & Data — Summer 2025 — Laney Strange

APP 7
Due: June 5th, 2025 @ 11:30am via Gradescope

Name:

- APPs will be assigned towards the end of roughly two lectures each week. You'll put together a solution to a short problem that we'll all use in the following lecture. We'll have time set aside to do these in class, or you can work on your own.
- You may handwrite your solutions, or typeset them in LaTeXor another system.
- APPs will be graded on completeness. They must be submitted by 11:30am (just before lecture) on the due date. They will not be accepted late, but we drop 3 of them (out of 8 total).
- Collaboration is strongly encouraged for APPs!

Problem 1. *DFS*

Topological sort only works on a directed, acyclic graph. A *back edge* is an edge (u, v) connecting a vertex u to an ancestor v in a depth-first tree. Self-loops are considered back-edges. We want to prove that:

- if a directed graph G is acyclic, then a DFS of G yields no back edges.
- if a DFS of a directed graph G yields no back edges then G is acyclic , and

The first part of the proof is below. Your job is to add the second part.

(Part 1) We will show that if a directed graph G is acyclic, then a DFS of G yields no back edges. Proof by contrapositive: show that if a DFS of G produces a back edge, then G has a cycle.

Suppose a DFS search produces a back edge (u, v). Then vertex v is an ancestor of u in the DFS forest. Thus G contains a path from v to u and the back edge (u, v) would complete a cycle.

Solution:
(Part 2) Show that if a DFS of a directed graph G yields no back edges, then G is acyclic. Proof by contrapositive: show that if G has a cycle, then a DFS on G yields at least one back edge.