
Homework 9: Q2

Name: Stephen Yang, syang29

*Don't forget to input your list of collaborators and sources on **AutoLab**.*
Please submit this file as a PDF.

1 Part (a): Algorithm Idea

In the adjacency graph, get the adjacency list for the given vertex and check that there is no outgoing edge to any other vertex. If there is an outgoing edge, return false.

If there is none continue to go through every vertex's adjacency list, and only check the index of the vertex given if it is connected to the vertex given. The first time that the vertex being checked in the loop does not have an outgoing edge to the given vertex index, return false. At the end return true.

2 Part (a): Runtime Analysis

To check one vertex's adjacency list takes $O(1)$ to access the list and then $O(n)$ for each item in the list.

To check one item in each vertex's adjacency list, assuming that adjacency graph is an array of arrays, accessing an array takes $O(1)$ and a specific location in an array takes $O(1)$. We will do that for all vertexes, thus, $O(n)$ times.

The first check thus takes $O(n)$, and the second takes $O(n)$ and thus the algorithm as a whole takes $O(n)$.

3 Part (b): Algorithm Idea

4 Part (b): Algorithm Details

Algorithm: Algorithm1

5 Part (b): Proof of Correctness Idea

6 Part (b): Proof Details

7 Part (b): Runtime Analysis