Homework 7: Graph Algorithms

Due: November 7, 2024 at 11:59 pm

Problem 1. Given an arbitrary undirected graph $G = (V, E)$, applying DFS on a given vertex will create a tree. The tree can be used to detect the separating edges and vertices of the graph.

First, some notation: Given $G = (V, E)$, a Depth-First Tree rooted at $r \in V$, and a pair of distinct vertices $u, v \in V$ such that there is a directed path comprised only of forward (discovery) edges of the Depth-First Tree from u to v, we say that u is an *ancestor* of v and v is a *descendant* of u. If u is an ancestor of v and the edge (u, v) is part of the Depth-Fist Tree then we say that u is a direct ancestor of v and v is a direct descendant of u.

(a) Show that the root vertex of a DFS tree for G is a separating vertex of G if and only if if it has more than one direct descendant in the DFS tree.

(b) Show that any non-root vertex v of a DFS tree is a separating vertex of G if and only there exists a child of v, w, such that none of w and w's descendants (if they exist) in the DFS tree have a back-edge to an ancestor of v in the DFS tree.

Solution. □

Problem 2. The city of Irvine, California, allows residents to own a maximum of three dogs per household without a breeder's license. Imagine you are running an online pet adoption website for the city for n Irvine residents and m puppies.

Describe an efficient algorithm for assigning puppies to residents that provides for the maximum number of puppy adoptions possible while satisfying the constraints that each resident will only adopt puppies that he or she likes and that no resident can adopt more than **three** puppies. Provide a proof of the correctness of your algorithm along with an analysis of runtime and space complexity.

Solution. □