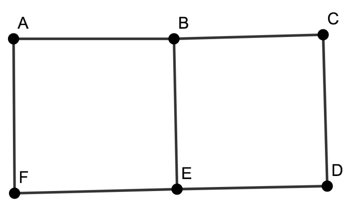
Assignment 21:
More on DFS
Due in class Thursday, 3/07

March 4, 2019
CS: DS&A
Proof School

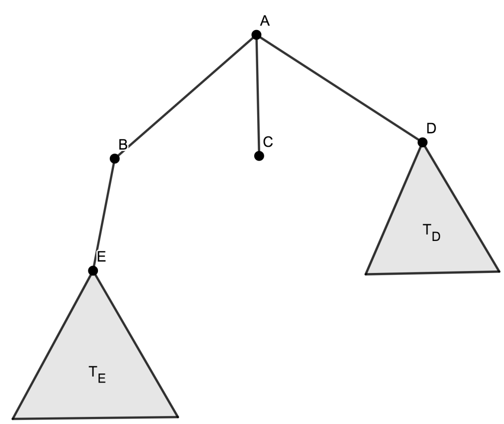
Problem 1. Draw as many different DFS trees as you can that result from doing DFS to the graphs below. (I consider two trees to be different if there's no 1-1 correspondence between them that preserves the root and the edges. Different node labels don't matter.)

a) K_5 (i.e. the complete graph on 5 vertices)

b)



Problem 2. Suppose we do DFS on a graph and get the following DFS tree. T_E and T_D are subtrees of nodes E and D.



a) What can you say about start/finish times of E compared to B?

b) What can you say about the start/finish times of nodes in T_E compared to start/finish times of nodes in T_D?

c) Recall that a *back edge* is any edge of the graph that's not recorded as a tree edge in the DFS tree. What nodes can E be connected to via a back edge? Suppose $v \in T_E$. What nodes can v be connected to via a back edge?

Problem 3. Describe an algorithm that uses DFS to determine whether a graph G has a cycle or not. Try to prove that your algorithm works as carefully as you can.

Problem 4. *(Optional and Extra Credit, but give it some thought!)*
Let G be a connected, undirected graph. Recall from the end of class that a *cut node* is a node v with the property that, when we remove v and all edges from v, the resulting graph is disconnected.

By working through examples and thinking about DFS trees and back edges, come up with an $\Theta(|V| + |E|)$ algorithm for finding all the cut nodes in G.