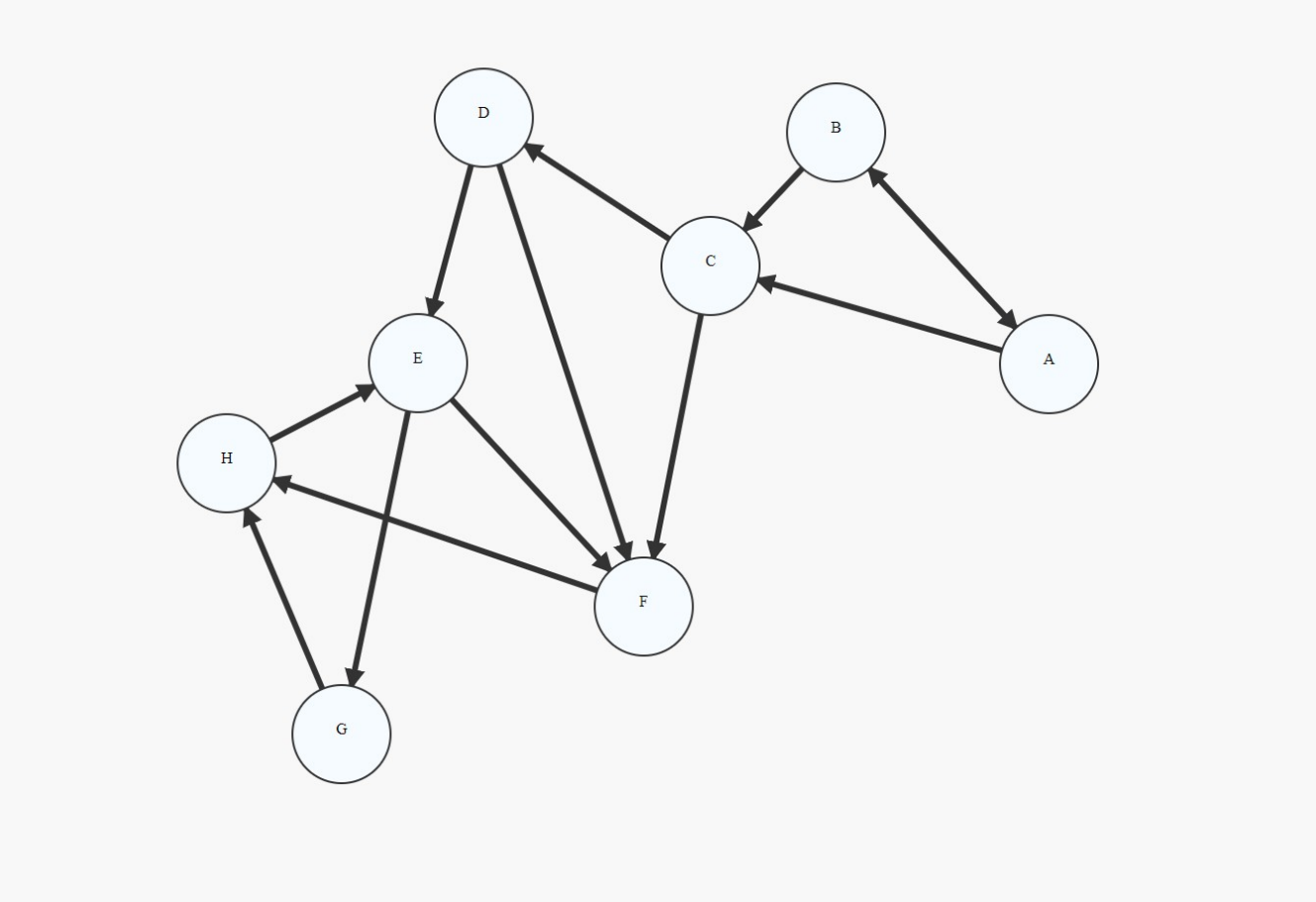
Project #3
Ryan English
04/11/21

I wrote an HTML/Javascript page that displays a graph and allows you to select the source node, and run the breadth first search and the depth first search; with the depth first search giving the extra edge information.

I utilized some pre-built HTML/Javascript code to build and display the graph, found here: [Interactive tool for creating directed graphs using d3.js. - bl.ocks.org]

The graph:



How to use:

1. Open indext.html

2. Select a source node

3. Click BFS / DFS

4. Watch the algorithm work

5. Click Reset to start a new graph

Note: If graph acts weird just refresh

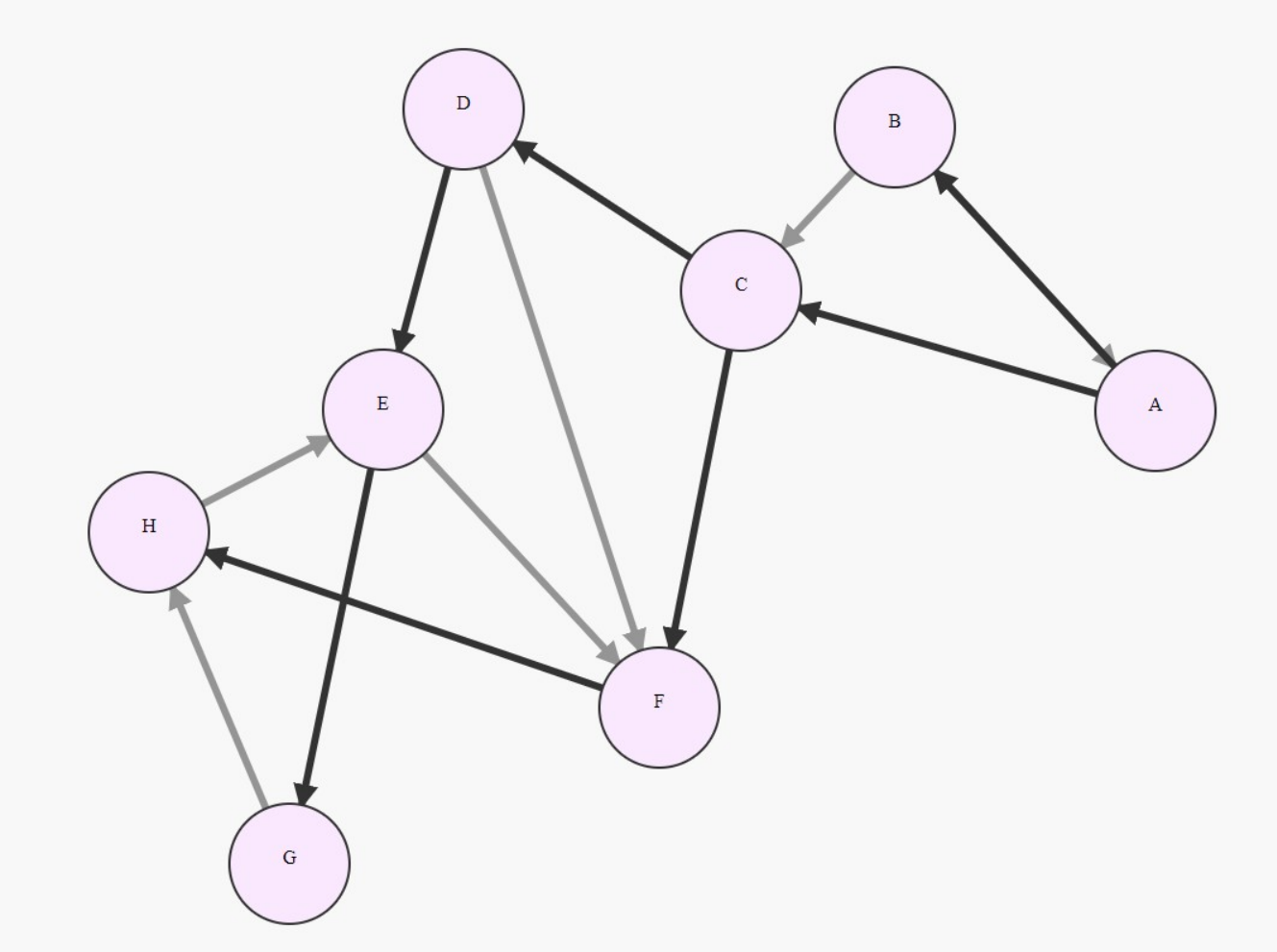
1) What is the BFS

Breadth first search utilizes a queue in order to go to all adjacent nodes first then traverse down; generating a “fat” minimum spanning tree.

Input: Graph G and Source node A

Output:

Light gray edges are not included



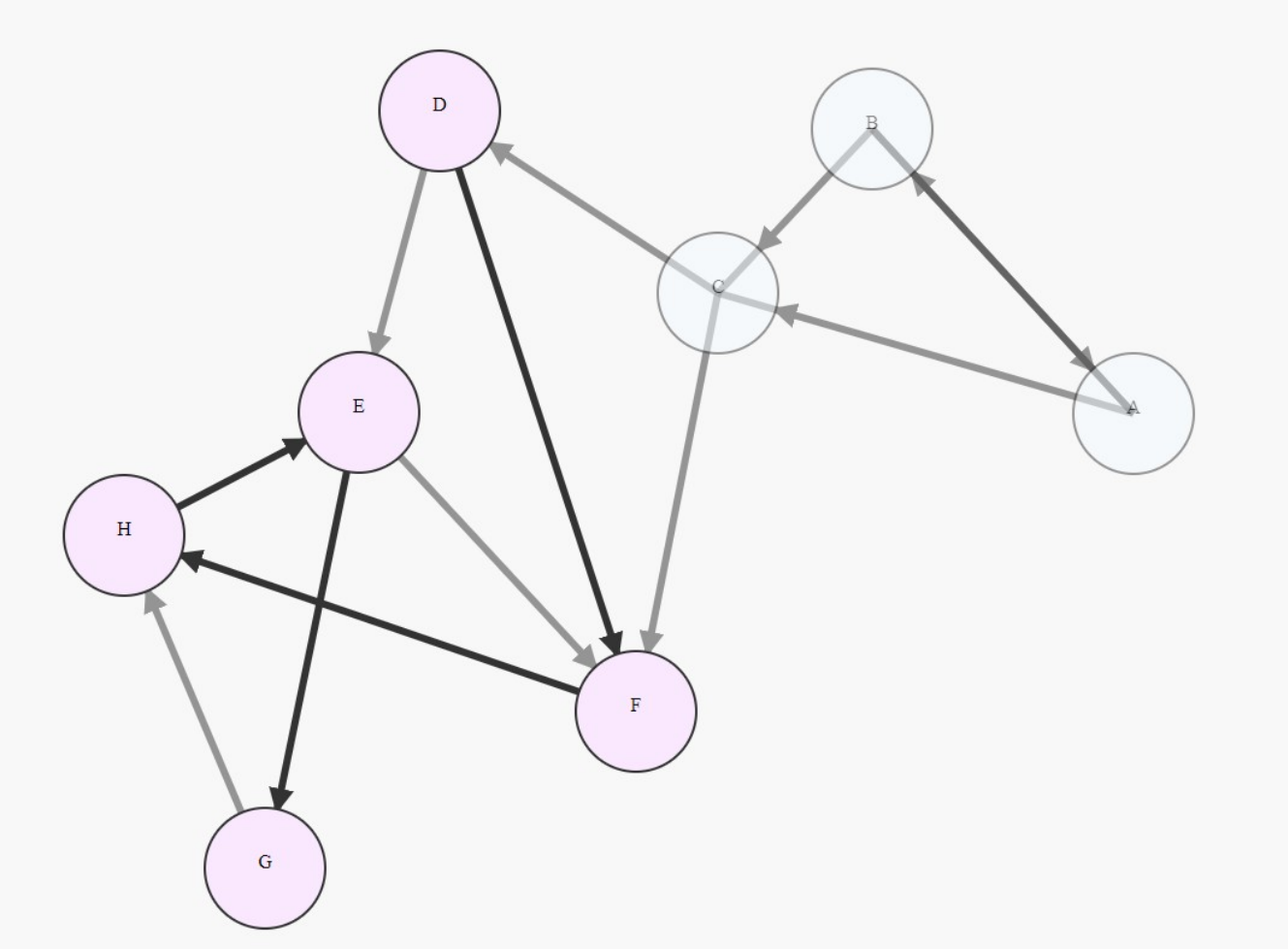
2) What is the DFS

Depth first search utilizes a stack to go deep into a branch before going to adjacencies; creating a thin graph/

Input: Graph G and Source node D

Output:

** Gray edges and half opacity nodes are not part of the graph **



3) Four edge types

Given the graph in number 2

Depth First Search

Tree Edges

- D -> F
- F -> H
- H -> E
- E -> G
- H -> E

Back Edges

Cross Edges

- E -> F

Forward Edges

- E -> F
- G -> H
- E -> G
- C -> D
- C -> F
- B -> A
- B -> C

Topological Sort

A,C,B,C,F,D,F,E,G,F,H,

Tree Edge (u, v)
Edges that are part of the DFS

Back Edge (u, v)
Edge that is explored and v is an ancestor of u; and u is grey at the time of exploring

Forward Edge (u, v)
Edge that is explored and distance of v is greater than distance of u

Cross Edge (u, v)
Edge that is explored and distance of u is greater than distance of v

4) If G is acyclic find its toplogical listing

The DFS in 2/3 does not have any back edges; therefore we can list it in topological order.

Topological Sort

A,C,B,C,F,D,F,E,G,F,H,