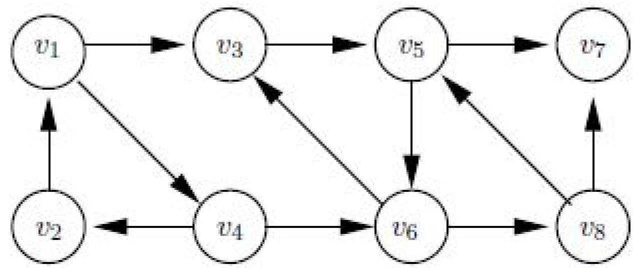
<u>Depth First Search:</u> (10 points each)

Answer each of the questions below by considering a run of DFS on the following graph. Each question may refer to a different DFS. **You do not need to provide justification.**



a) Given that v_8v_5 is a tree edge, list all edges that cannot be a tree edge.

- v3v5 cannot be a tree edge. In depth-first search, a vertex can be discovered by at most one edge.
- v1v3 also can not be tree edge
- Grading will be: Full credit if they get both v1v3 and v3v5, (-2) points if they only get one of them

b) Given that $v_6.d = 1$ and $v_3.d = 2$, list all descendants of v_3 (i.e., list all u such that $v_3.d < u.d < u.f < v_3.f$).

- v5 and v7 are the descendants of v3, since there is a white path from v3 to these vertices at time v3.d.

c) Given that $v_4.d < v_3.d$, what is the relationship between $v_4.f$ and $v_3.f$?

- v4.f > v3.f . v6 must be white at time v4.d since v3 would have been discovered before v4 otherwise, thus there is a white path from v4 to v3 at time v4.d.

d) List all edges that definitely cannot be back edges.

- v1v3, v4v6, v5v7, v8v7. These are the edges that do not belong to a cycle.

e) List all vertices that can possibly have the largest finishing time.

- v1, v2, and v4. The strongly connected component that is composed of these vertices is the first vertex in a topological sort of the component graph.

f) What are the strongly connected components of this graph?

- C1: v1, v2, v4 C2: v3, v5, v6, v8   C3: v7

g) List all outgoing edges of v1 that can possibly be forward edges.

- v1v3   (v1v4 must be tree edge since v4 can only be discovered through v1v4)

h) Given that v2.d = 1, what is v2.f?

- V2.f = 16

Greedy Choice Counter-Example (14 Points)

1) Consider the following variant of the activity selection problem: We are given n intervals and a weight w_i for each interval i. We would like to determine the set of non-overlapping activities with the largest total weight. Provide a counter-example to show that the following greedy algorithm is not guaranteed to find the optimal solution for this problem: Take the activity with the largest weight, remove all activities overlapping with this activity, and repeat until no activity is left.
   - Consider the following instance with three activities:
     s1 = 1, f1 = 3, w1 = 2
     s2 = 2, f2 = 5, w2 = 3
     s3 = 4, f3 = 6, w3 = 2
     The optimal solution for this instance is {s1, s3} with value 4. The proposed greedy algorithm returns the solution
     {s2}, with value 3.

2) Consider the following variant of the activity selection problem: We are given n intervals. We would like to determine the set of non-overlapping activities with the longest total duration. Provide a counter-example to show that the following greedy algorithm is not guaranteed to find the optimal solution for this problem: Take the activity with the longest duration, remove all activities overlapping with this activity, and repeat until no activity is left.
   - Consider the following instance with three activities:
     s1 = 1, f1 = 3
     s2 = 2, f2 = 5
     s3 = 4, f3 = 6
     The optimal solution for this instance is {s1, s3} with value 4. The proposed greedy algorithm returns the solution
     {s2}, with value 3.

3) Provide a counter-example to show that the following greedy algorithm is not guaranteed to find the optimal solution for the activity selection problem: Take the activity with the shortest duration, remove all activities overlapping with this activity, and repeat until no activity is left.

   - Consider the following instance with three activities:
     s1 = 1, f1 = 4
     s2 = 3, f2 = 5
     s3 = 4, f3 = 7

The optimal solution for this instance is {s1, s3} with value 2. The proposed greedy algorithm returns the solution {s2}, with value 1.

Interview Questions: (18 points each)

1) We are given a city map in which all roads are one-way. We say a map is "complete" if it is possible to reach from any intersection to every other intersection in the city. Which algorithm should we use to figure out whether a given map is complete?
   - We can create a graph in which vertices represent intersections and directed edges represent one-way roads. Then we can find the strongly connected components. If the graph has only one strongly connected component, then the map is complete.
   -

2) Why are the strongly connected components of a directed graph disjoint?
   - If the components are not disjoint then this would violate the definition of strongly connected component. If there is at least one vertex in the intersection of Ci and Cj, then every vertex in Ci (Cj) can reach every vertex in Cj (Ci) through that vertex. In this case, Union(Ci, Cj) would be strongly connected, thus Ci and Cj would not be maximal.

3) Why is the component graph of any directed graph acyclic?
   - If there is a cycle in the component graph, then all vertices in each of the components involved in the cycle would be able to reach each other. In this case, the union of the components in the cycle would be strongly connected, thus these components would not be maximal, violating the definition of a strongly connected component.

4) How many strongly connected components are there in a directed acyclic graph G = (V,E)? Why?
   - If there is no cycle in the graph, then no pair of vertices can reach each other. Therefore, there will be |V| strongly connected components in the graph.

5) Let H be the component graph of a directed graph G. What is the component graph of the transpose of G? Why?
   - The component graph of the transpose of G will be the transpose of H. Since the SCCs of G and its transpose are identical, the vertices of the transpose will be the same as the vertices of H. However, all edges between the components will be flipped.

Greedy Algorithms (20 pts):

You are planning your road trip on the highway. You know that there are *n* gas stations on your way, at miles $0 = s_0 < s_1 < s_2 < ... < s_n = m$ and you can go *k* miles when you start with a full tank. You would like to minimize the number of stops you make, while making sure that you will never run out of gas. For example, if $s = <0, 90, 180, 250, 270, 310, 350, 500>$, and $k = 200$, then an optimal solution is to stop at stations s_2 (mile 180) and s_5 (mile 310).

State the greedy choice property for this problem. (no need to prove)

- Greedy choice property: Let s_j be the last station before mile k. There is an optimal set of stations that contains s_j.