Homework 3b Solutions
CSI 503 Spring 2021

Rahul Bhenjalia

April 5 2021

1 Problems

1. Consider two representations of a directed graph: adjacency matrix and adjacency list. For the following tasks, determine their tight Big-O time complexity when each of the representations are used. Provide one or two lines of justification for each answer.

   For adjacency list, we have each node maintaining a list of all its adjacent edges. Let's assume that there are V number of nodes and E number of edges in the graph.

   For adjacency matrix, we have a VxV array.

   (a) Calculating in-degree of vertex u.

   ANSWER: Given the adjacency list representation, the in-degree of a vertex u is equal to the number of times it appears in all the list of given Adj list. If we search all the lists for each vertex, time to compute the in-degree of all the vertices would be $O(VxE)$
   . **The adjacency matrix of any graph has** $O(V^2)$ **entries, regardless of number of edges in the graph. Hence the computing the in-degree of a vertex u is equivalent to scanning the column corresponding to u and summing the ones, thus time required is also** $O(V)$ **for one vertex.**

   (b) Calculating the transpose of the graph.

   ANSWER: The algorithm for transpose would generally iterate over all edges and reverse their direction. Hence, for adjacency list it would be $O(V+E)$ **and for adjacency matrix it would be** $O(V^2)$**.**

   (c) Depth-First Search (consider the pseudocode given in the textbook).

   ANSWER:
   . **If the graph is represented as adjacency list: For each node, we discover all its neighbours by traversing the list, just once in linear time. Hence, for a directed graph, the sum of the sizes of adjacency lists of nodes is E. So, the time complexity in this case is**

$O(V+E)$.

. **If the graph is represented as adjacency matrix: For each node, we will have to traverse an entire row of length V in the matrix to discover all its outgoing edges. Also, each row in an adjacency matrix corresponds to a node in the graph, and that row stores information about edges emerging from the node. Hence, the time complexity of DFS in this case is is** $O(V^2)$.

2. Consider the directed graph in Figure 24.8 in CLRS (Page 667). The graph has 6 vertices (v0, . . . , v5). Your task is to perform graph traversal subjected to the following condition: When you have multiple choices for visit (e.g., in a loop over all vertices or adjacent edges), choose the vertex or the edge with the smallest vertex index. For example, if you have a choice between visiting vertices v2, v4, and v5, you must first visit v2, then v4, and finally v5. Note that this restriction should lead you to a unique response if you follow the algorithms correctly (no randomness involved).

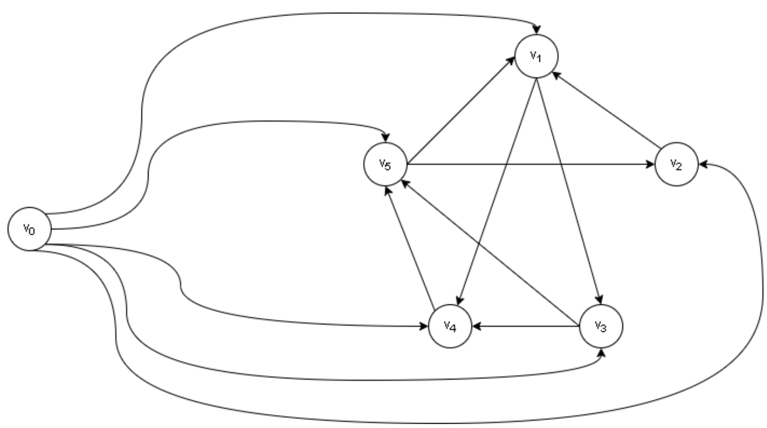

Figure 1: The Given Graph for traversal without edge weights and vertex values

(a) Perform BFS starting from v0 and mark the discovery time of each vertex (time starts from 1).

ANSWER: Since, we can see from the given graph that all the other nodes v_1, v_2, v_3, v_4, v_5 **are adjacent nodes of** v_0**, the discovery time of all of them would be the same in BFS.**

. **B**FS = $v_0, v_1, v_2, v_3, v_4, v_5$
. **D**iscovery Time = $1, 2, 2, 2, 2, 2$

(b) Perform DFS and mark the discovery and finishing time of each vertex (time starts from 1).

ANSWER:

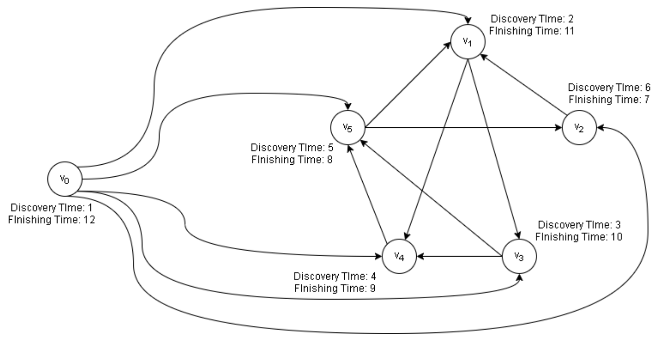

Figure 2: Discovery and Finishing Times in DFS

. **D**FS = $v_0, v_1, v_3, v_4, v_5, v_2$
. **D**iscovery Time = $1, 2, 3, 4, 5, 6$
. **F**inishing Time = $12, 11, 10, 9, 8, 7$

(c) Classify and label each edge of the depth-first forest into tree (T), back (B), forward (F), and cross (C).

ANSWER:

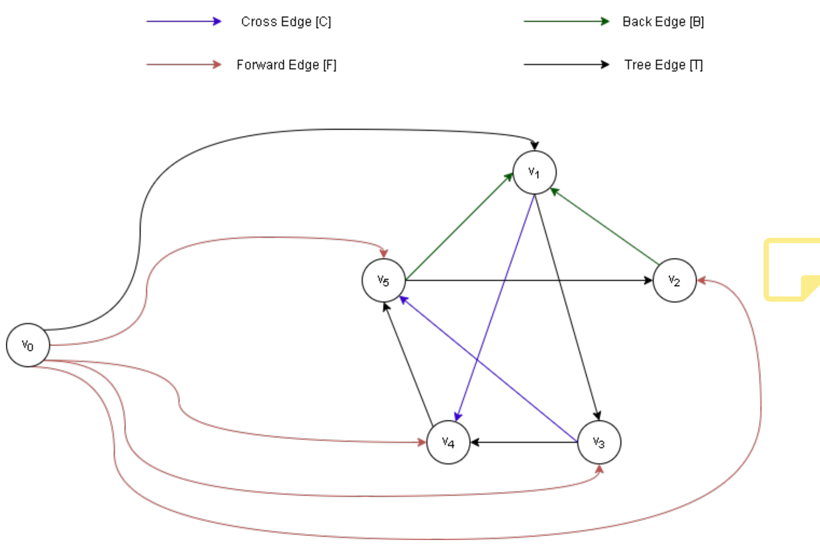

Figure 3: Edge Labels

(d) How many strongly connected components does the graph have? Determine the vertices in each component. Draw its component graph.

ANSWER: There are total of 2 Strongly connected components and they are as follows:

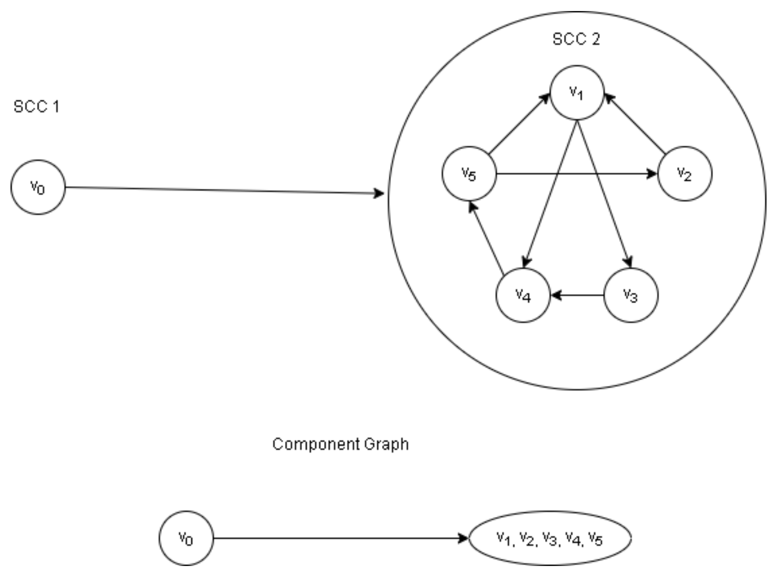

Figure 4: Component Graph