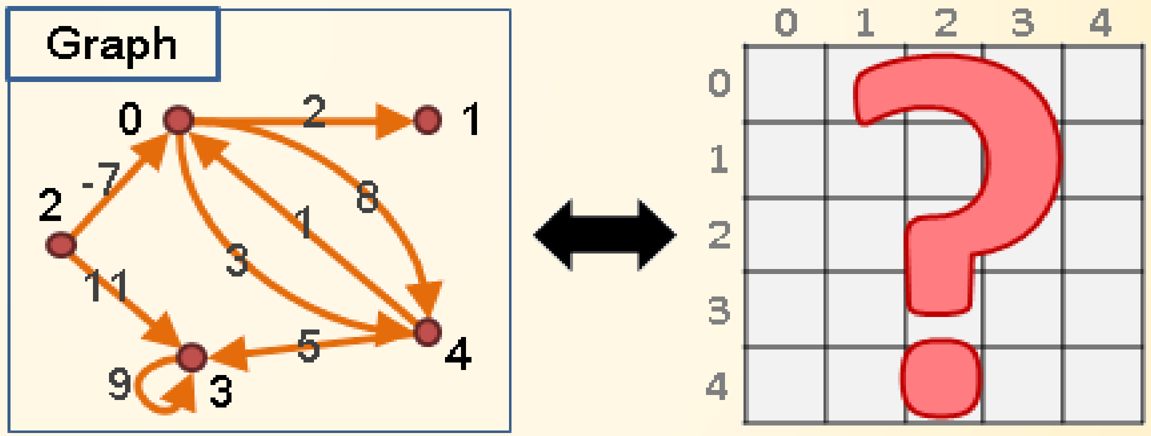
Ch.6.1-6.3 Graph 參考答案

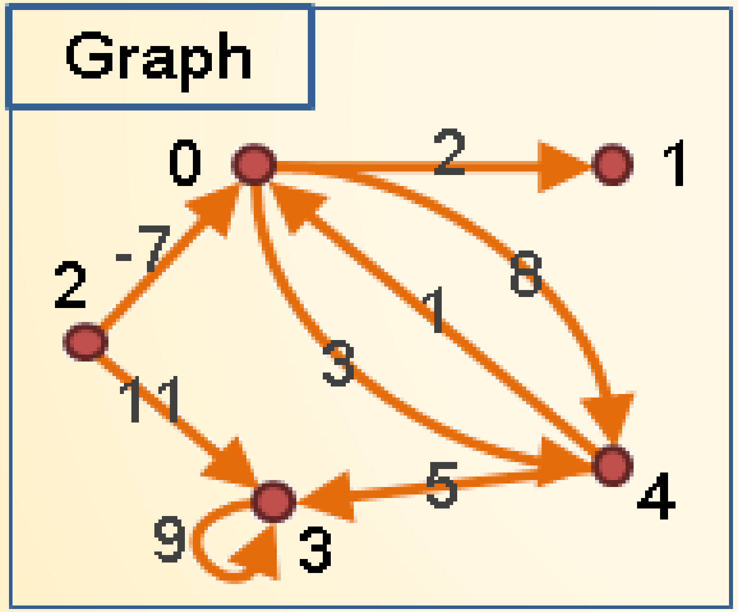
Question 1

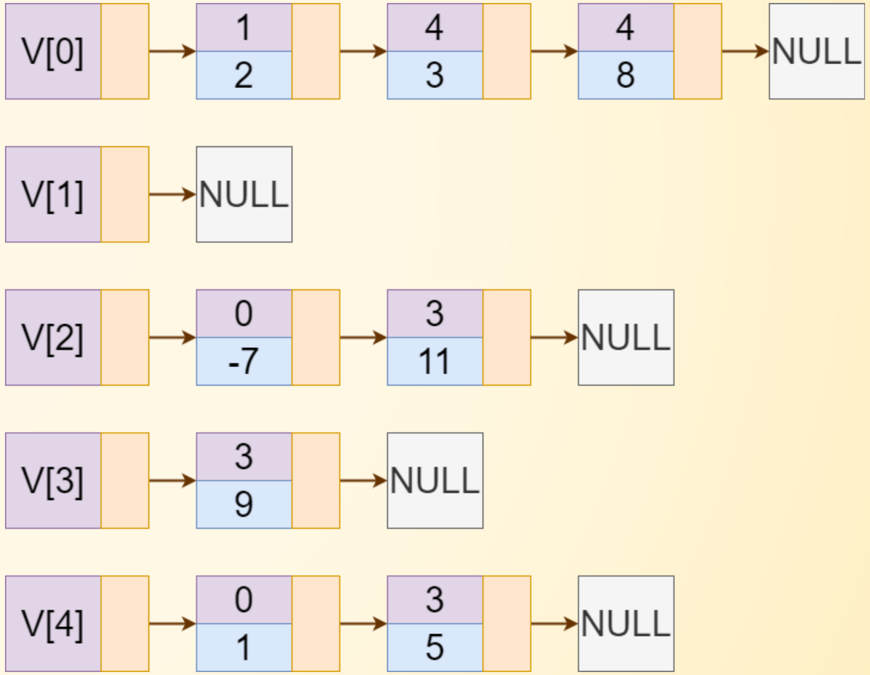
Can you use an adjacency matrix to represent the following graph? If yes, show your adjacency matrix; otherwise, show how you can represent the graph.



Answer 1

NO! there are two weighted edges from node “0” to node “4”.

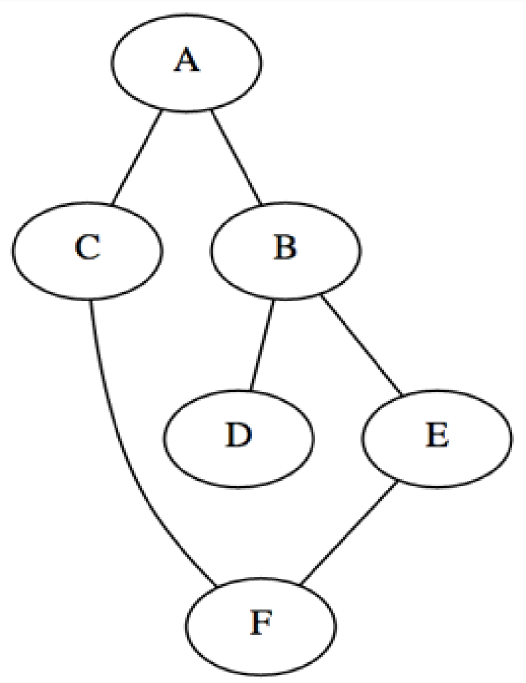




Question 2

(a) Explain the concept of BFS and DFS, and show the BFS and DFS search sequence of the following graph.

(b) Can you use the BFS to search a Tree data structure? If so, what is the equivalent tree traversal algorithm.



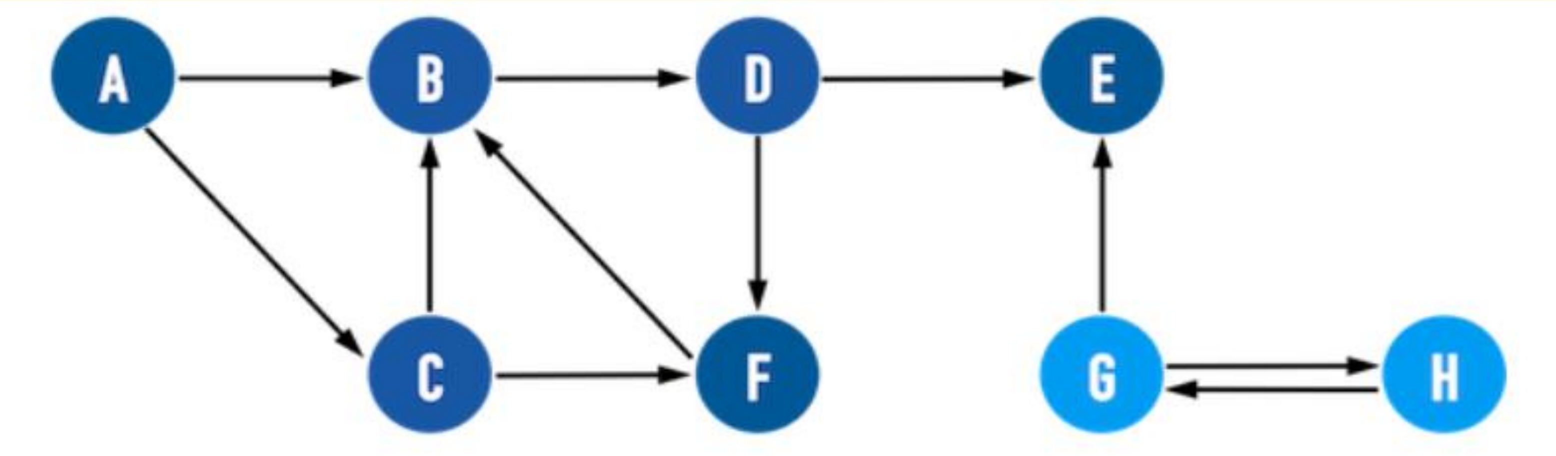
Answer 2

(a) as described in the class video.

(b) level-order traversal

Question 3

Please propose a method to determine whether a graph is cyclic, i.e. there exist cycles in the graph. You may try to adapt the DFS algorithm to solve this problem.

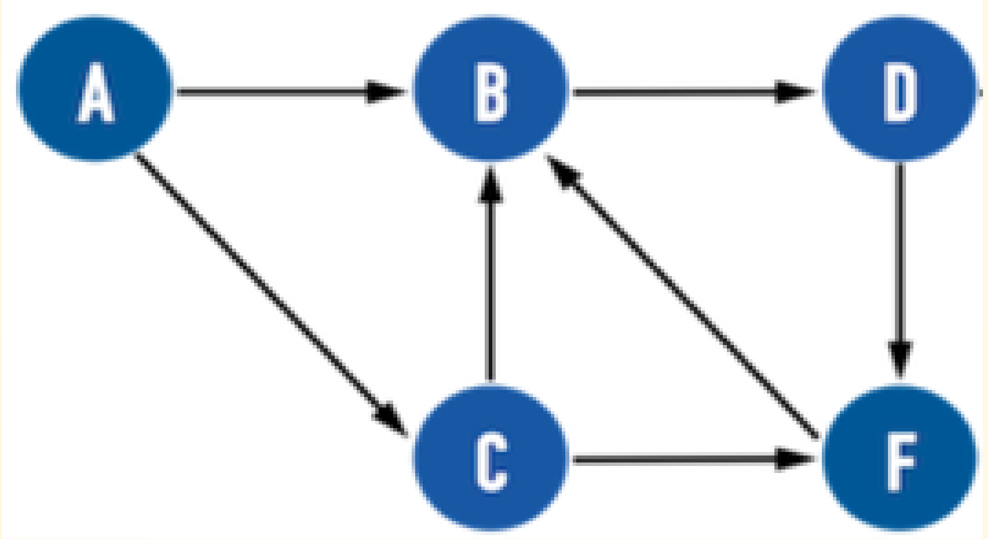


Answer 3

Depth First Traversal can be used to detect a cycle in a Graph. DFS for a connected graph produces a tree. There is a cycle in a graph only if there is a back edge present in the graph. A back edge is an edge that is from a node to itself (self-loop) or one of its ancestors in the tree produced by DFS.

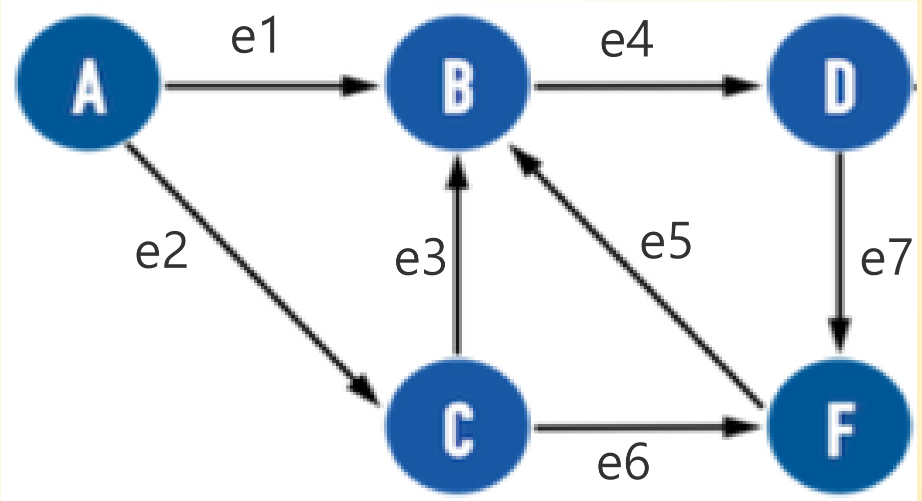
Question 4

Please represent the following graph using an incidence matrix.



Answer 4

	e1	e2	e3	e4	e5	e6	e7
A	1	1	0	0	0	0	0
B	-1	0	-1	1	-1	0	0
C	0	-1	1	0	0	1	0
D	0	0	0	-1	0	0	1
F	0	0	0	0	1	-1	-1



Question 5

Prove that the minimum weight edge of a graph must be included in the MST if every edge of the graph is of different weight.

Answer 5

(Proof by contradiction)

Let's assume that's not true, i.e., there exists a vertex v such that MST does not use any of its smallest weight edges (there may be more than one). Let e be any of such edges, then you can add e to MST and then remove the other edge of v on that cycle, which by definition was of strictly greater weight. We reach a contradiction with the weight of MST.

Question 6

Describe a method of the Prim's algorithm such that the time complexity is $O(V \log V + E \log V)$ or $O(E \log V)$.

Answer 6

Use a priority queue (implemented by a heap)

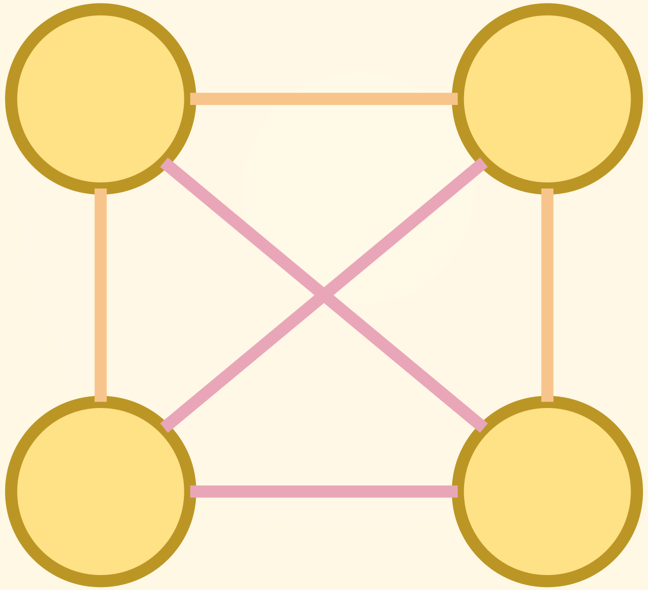
- ExtractMin : from all those vertices which have not yet been included in MST, we need to get vertex with minimum key value.
- DecreaseKey : After extracting vertex we need to update keys of its adjacent vertices, and if new key is smaller, then update that in data structure.

Question 7

Must any two spanning trees of a connected undirected graph have at least one common edge?

Answer 7

False.



Question 8

Graph edges can be divided into four categories after we apply DFS. Please find out their definitions respectively.

- Tree edge
- Back edge
- Forward edge
- Cross edge

Answer 8

- **Tree Edge:** It is an edge which is present in the tree obtained after applying DFS on the graph. All the Green edges are tree edges.
- **Forward Edge:** It is an edge <u, v> such that v is descendant but not part of the DFS tree. Edge from **1 to 8** is a forward edge.
- **Back edge:** It is an edge <u, v> such that v is ancestor of edge u but not part of DFS tree. Edge from **6 to 2** is a back edge. **Presence of back edge indicates a cycle in directed graph.**
- **Cross Edge:** It is a edge which connects two node such that they do not have any ancestor and a descendant relationship between them. Edge from node **5 to 4** is cross edge.

Question 9

An **articulation vertex** of a connected graph is a **vertex** whose removal will disconnect the graph.

(a) Describe how to use the DFS to find all the articulation vertices in a graph.

(b) Estimate its time complexity.

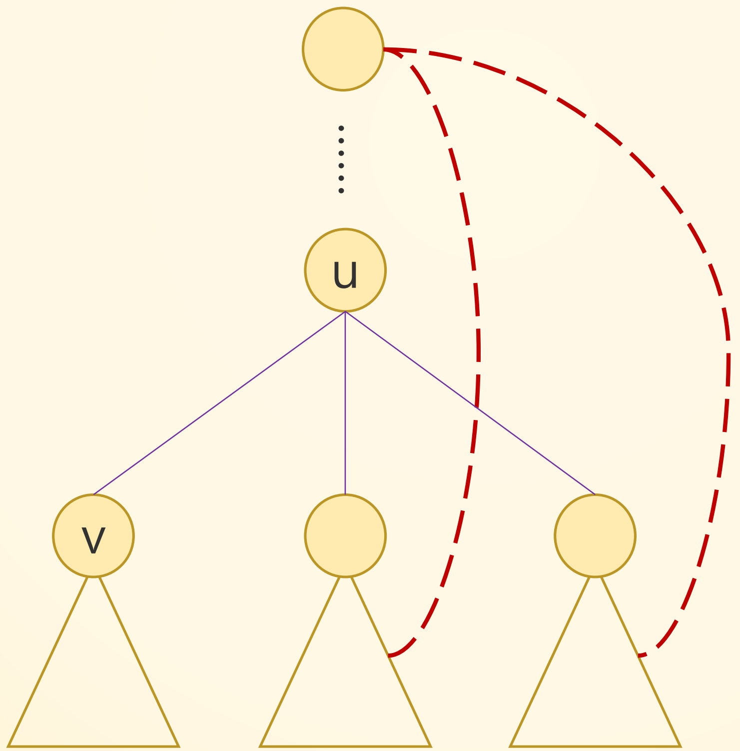
Answer 9

(a) Vertices after DFS form a tree called DFS tree. In a DFS tree, a vertex u is the parent of another vertex v, if v is discovered by u (obviously v is an adjacent of u in graph). In DFS tree, a vertex u is articulation point if one of the following two conditions is true.
(1) u is the root of DFS tree and it has at least two children.
(2) u is not the root of DFS tree and it has a child v such that no vertex in subtree rooted with v has a back edge to one of the ancestors (in DFS tree) of u.

(b) O(V+E)

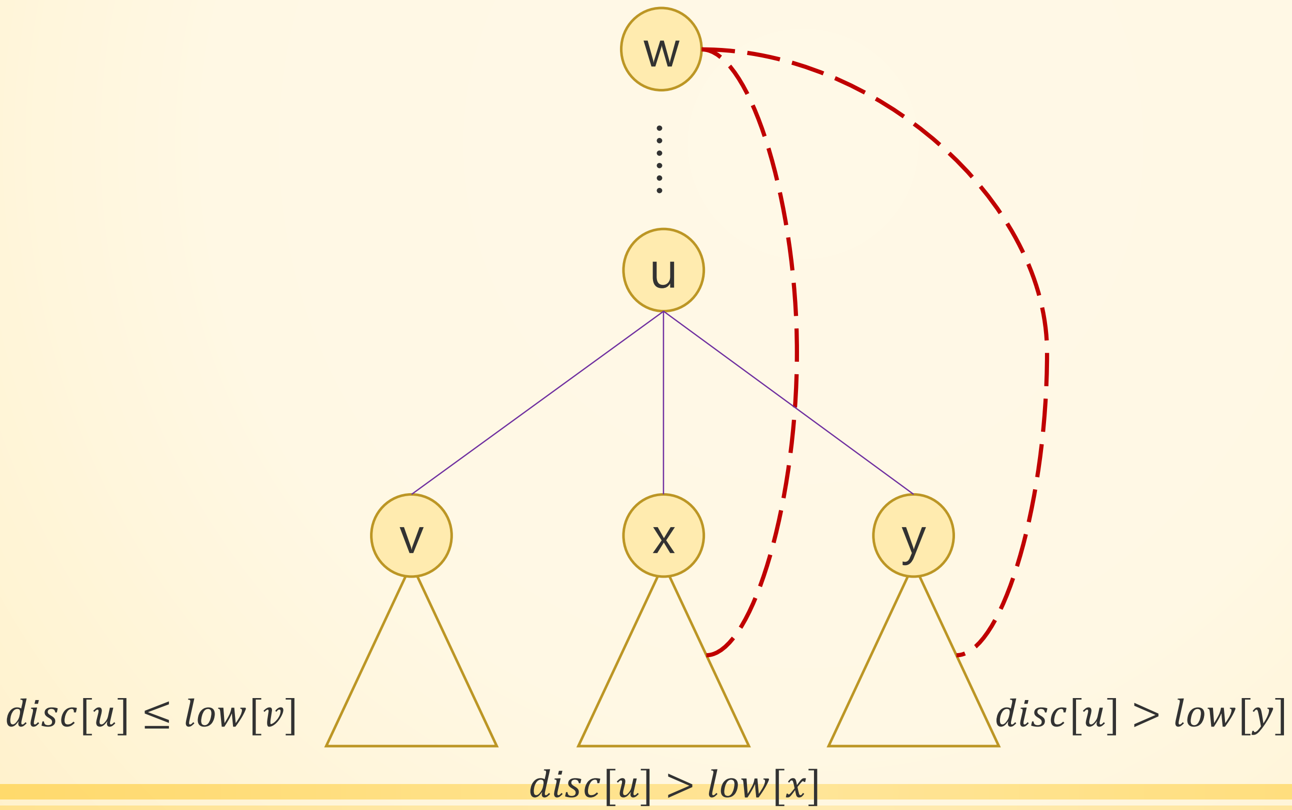
Answer 9 (Details)

u has a child v such that no vertex in subtree rooted with v has a back edge to one of the ancestors (in DFS tree) of u.



Answer 9 (Details)

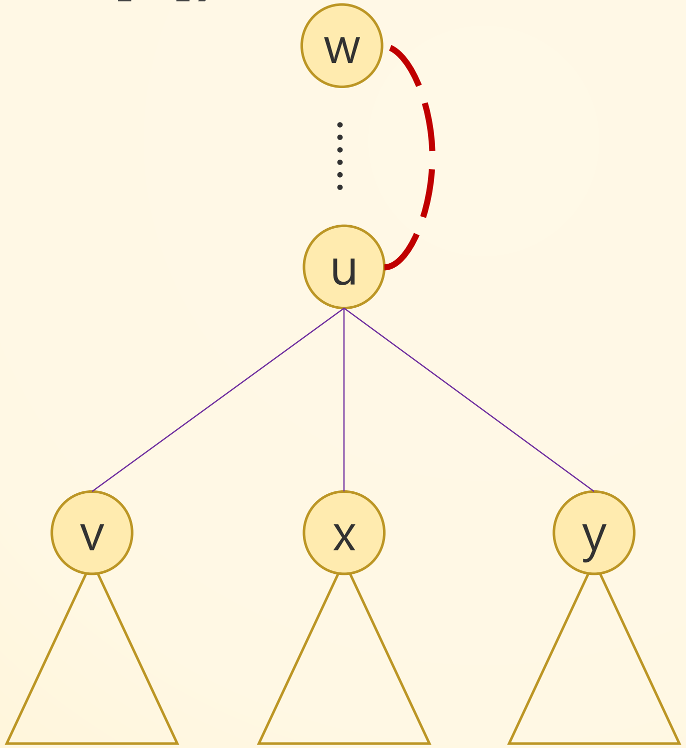
low[u] = min(disc[u], disc[w]) where w is an ancestor of u and there is a back edge from some descendant of u to w.



Answer 9 (Details)

How to update low[u] ?

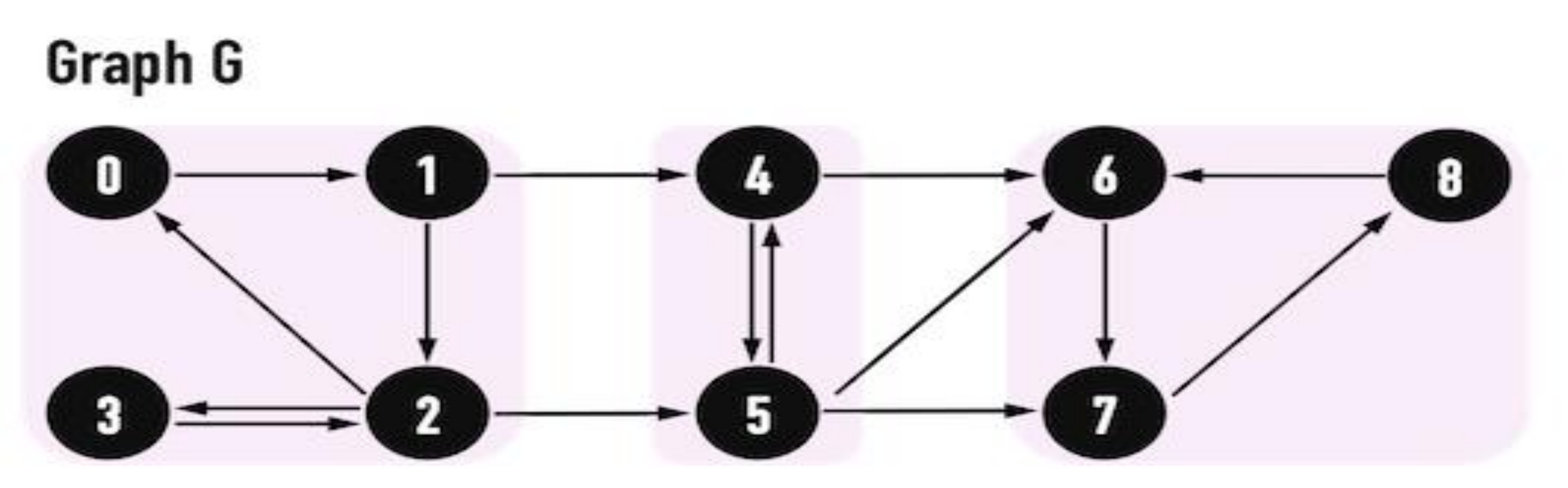
low[u] = min(low[u], disc[w])



Question 10

(a) Describe the definition of **strongly connected components (SCCs)**,

(b) Apply DFS twice to find out the SCCs of the following directed graph.



Answer 10

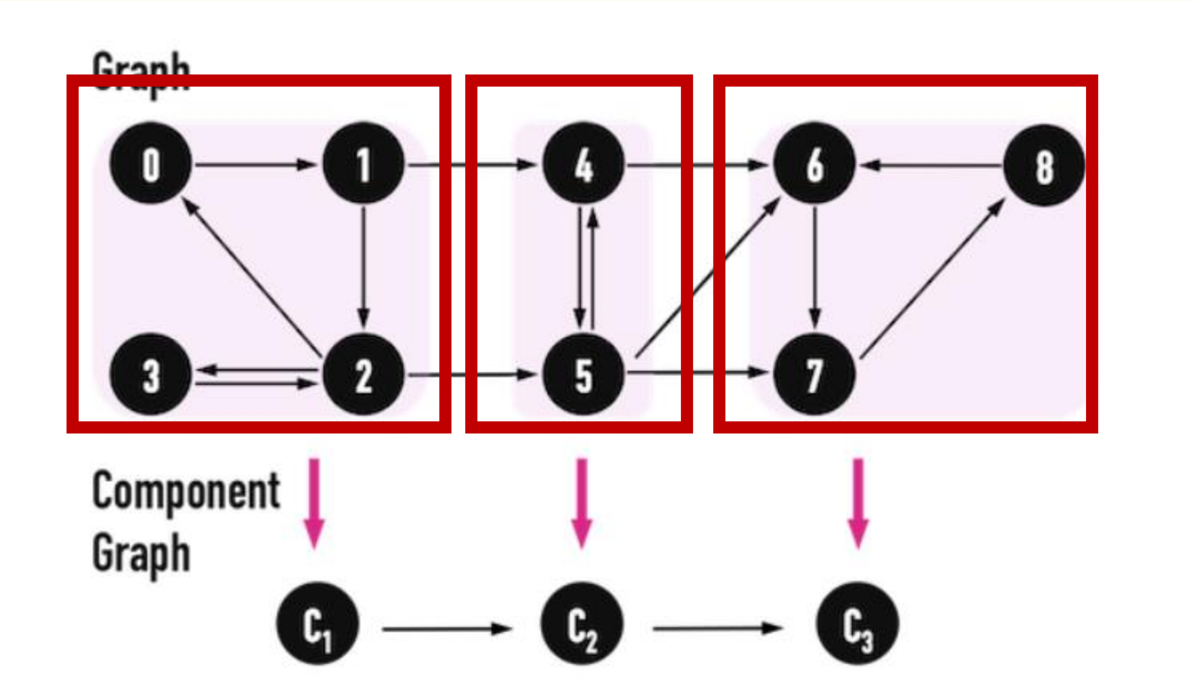
(a) A directed graph is strongly connected if there is at least a path between all pairs of vertices. A strongly connected component (**SCC**) of a directed graph is a maximal strongly connected subgraph. There could be more then one SCC in a graph.

(b) Step:

1. **Call** DFS(G) to compute finishing times f[u] for all *u*.
2. **Compute** G^T (G^T is G with all edges reversed)
3. **Call** DFS(G^T), but in the main loop, consider vertices in order of decreasing f[*u*] (as computed in first DFS)
4. **Output** the vertices in each tree of the depth-first forest formed in second DFS as a separate SCC.

Answer 10 (Details)

Why do we need to transpose G in step2 ?



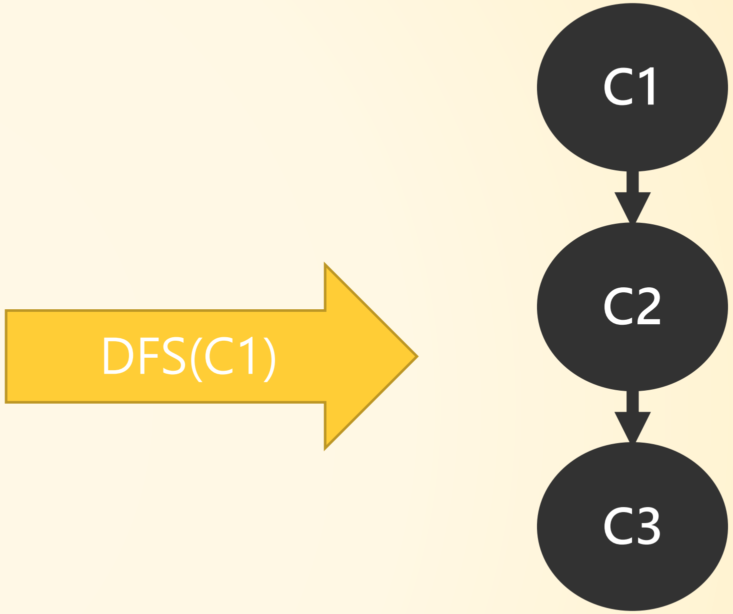
Answer 10 (Details)

Why do we need to transpose G in step2 ?

G:

Finish time: C1 > C2 > C3

Topological Sort



G^T:

Prove

Answer 10 (Reference for SCC)

- http://alrightchiu.github.io/SecondRound/graph-li-yong-dfsxun-zhao-strongly-connected-componentscc.html
- https://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/strongComponent.htm

Question 11

- Please implement Kruskal's algorithms to find the minimum spanning tree and its cost on the given graph

Answer 11

Please refer to:

https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-using-stl-in-c