

COL106: Data Structures and Algorithms

Lecture 33

Keerti Choudhary
Department of Computer Science, IIT Delhi

Announcements

1. Regrade requests for Minor 2, Quiz 2 and Quiz 3 must be submitted by April 08 (Saturday)
2. Make-up labs:
April 08 (Saturday) - Tuesday's group
April 15 (Saturday) - Friday's group

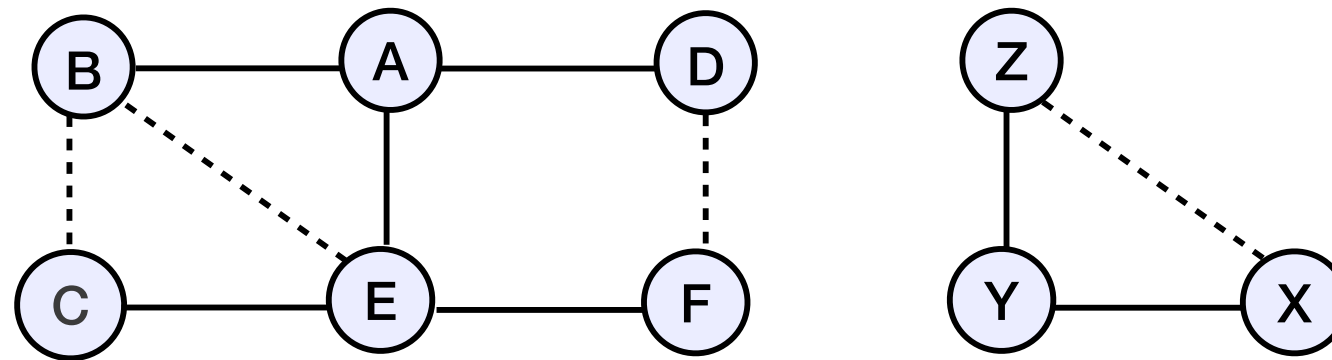
Today's Lecture

1. Quiz
2. Problem sheet-3 discussion

Question 1

Prove that any n vertex undirected graph G contains at most $n - 1$ bridge-edges. Also present examples to show that this bound is tight.

Part 1: Compute spanning tree of all connected components of G .



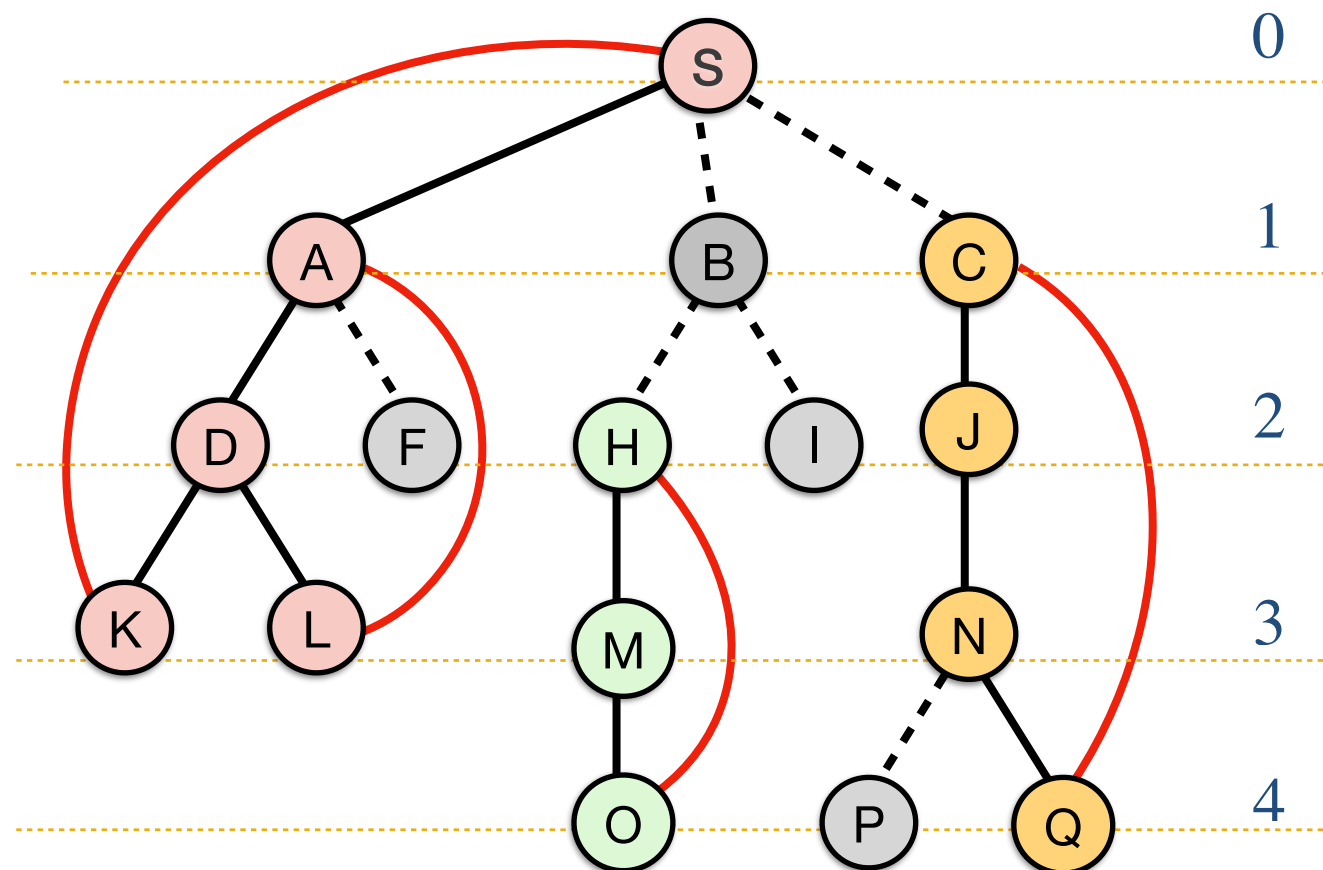
All bridge edges must lie in the resultant spanning forest, and so their count is at most $n - 1$.

Part 2: This bound is tight when G is a tree, and in such a case each of the edges of G is a bridge.

Question 2

Let $G = (V, E)$ be an undirected graph. Find a partition $\mathcal{P} = (V_1, \dots, V_k)$ of V in $O(m + n)$ time so that any pair $(x, y) \in V \times V$ is **bi-connected** iff they lie in same partition.

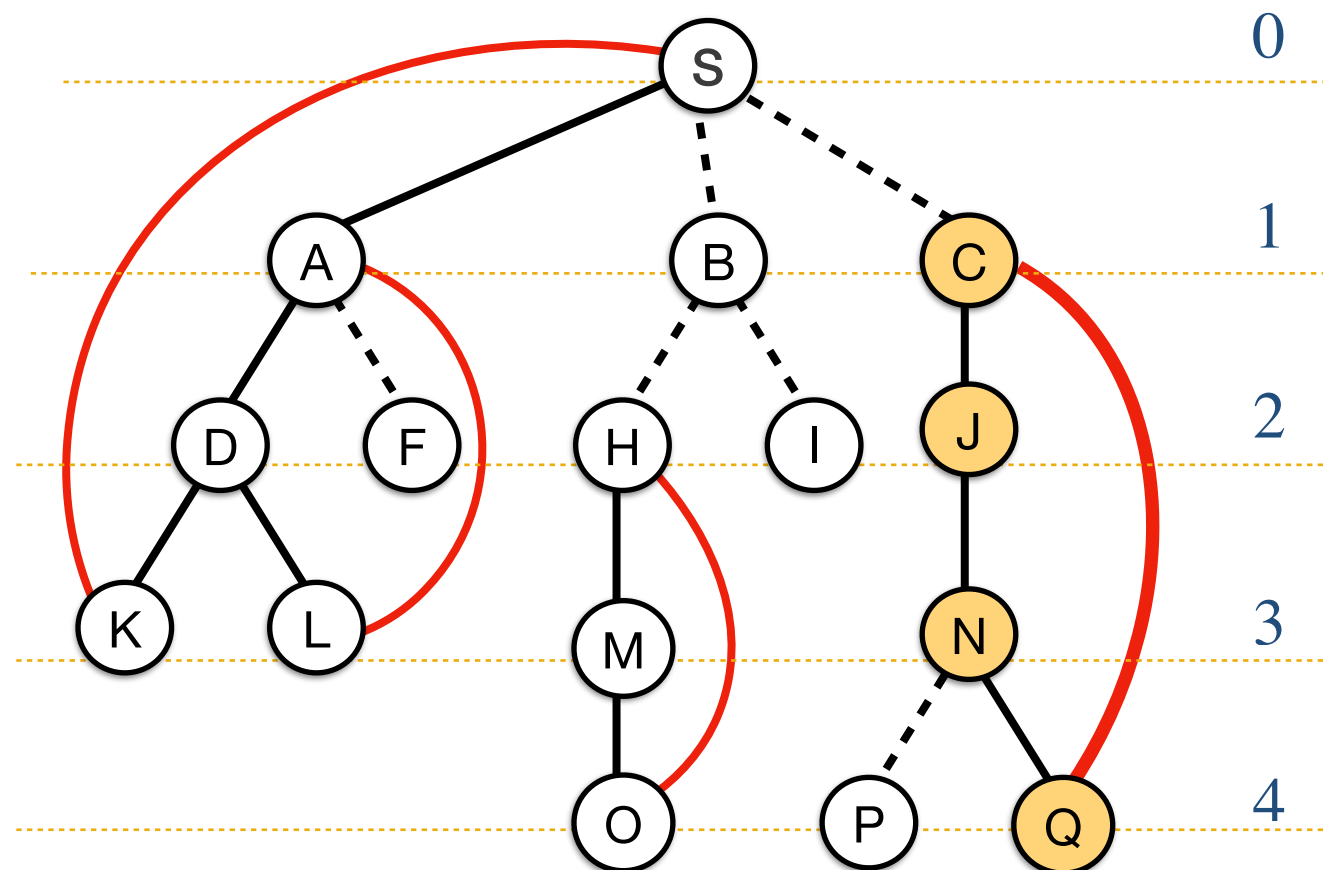
Lemma: Two vertices x, y are **bi-connected** if and only if the path from x to y in the DFS tree does not contain any bridge edges.



Question 2

Let $G = (V, E)$ be an undirected graph. Find a partition $\mathcal{P} = (V_1, \dots, V_k)$ of V in $O(m + n)$ time so that any pair $(x, y) \in V \times V$ is bi-connected iff they lie in same partition.

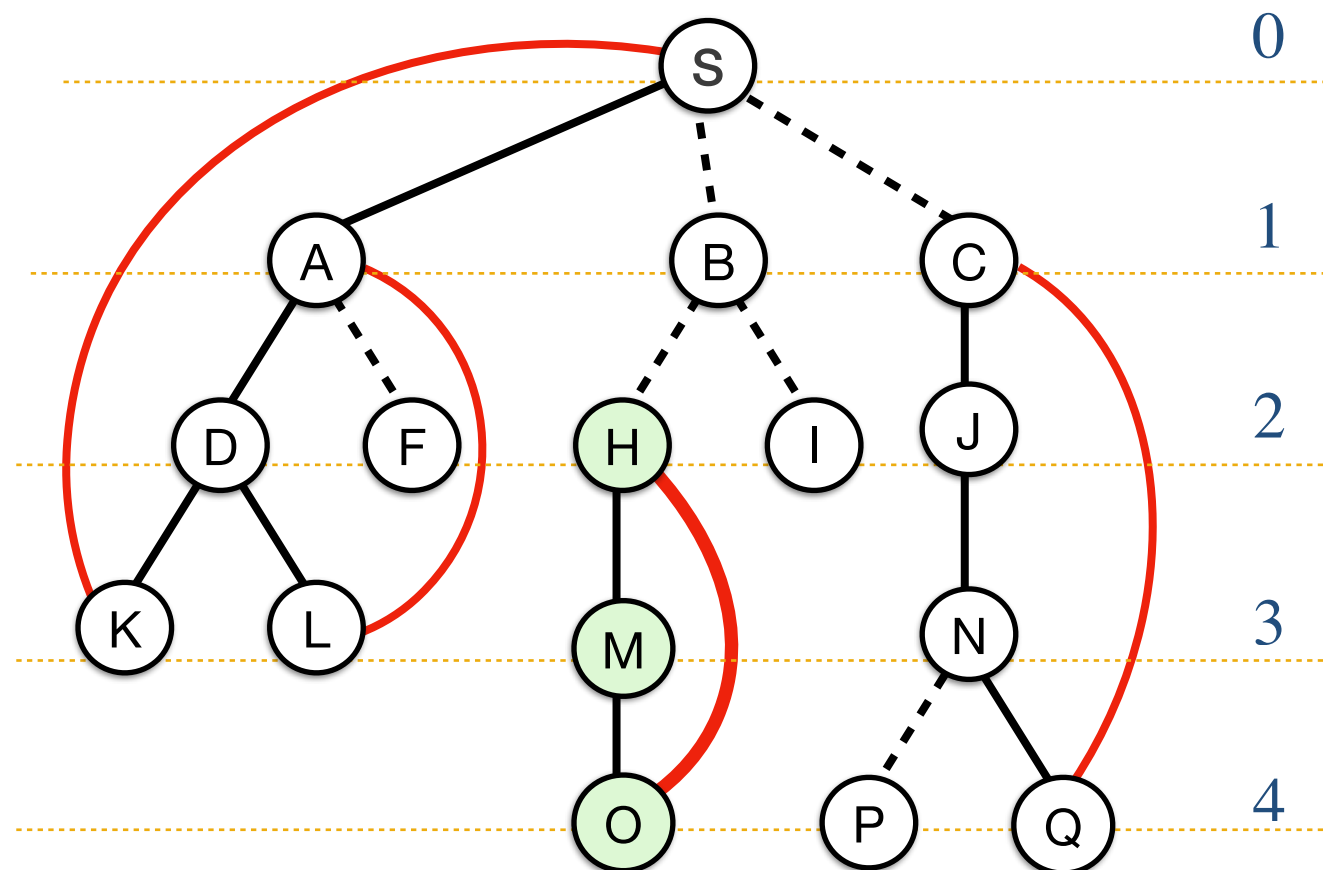
Claim 1: If (x, y) is a non-tree edge of DFS tree T , then all the vertices on cycle obtained by adding (x, y) to T are bi-connected.



Question 2

Let $G = (V, E)$ be an undirected graph. Find a partition $\mathcal{P} = (V_1, \dots, V_k)$ of V in $O(m + n)$ time so that any pair $(x, y) \in V \times V$ is bi-connected iff they lie in same partition.

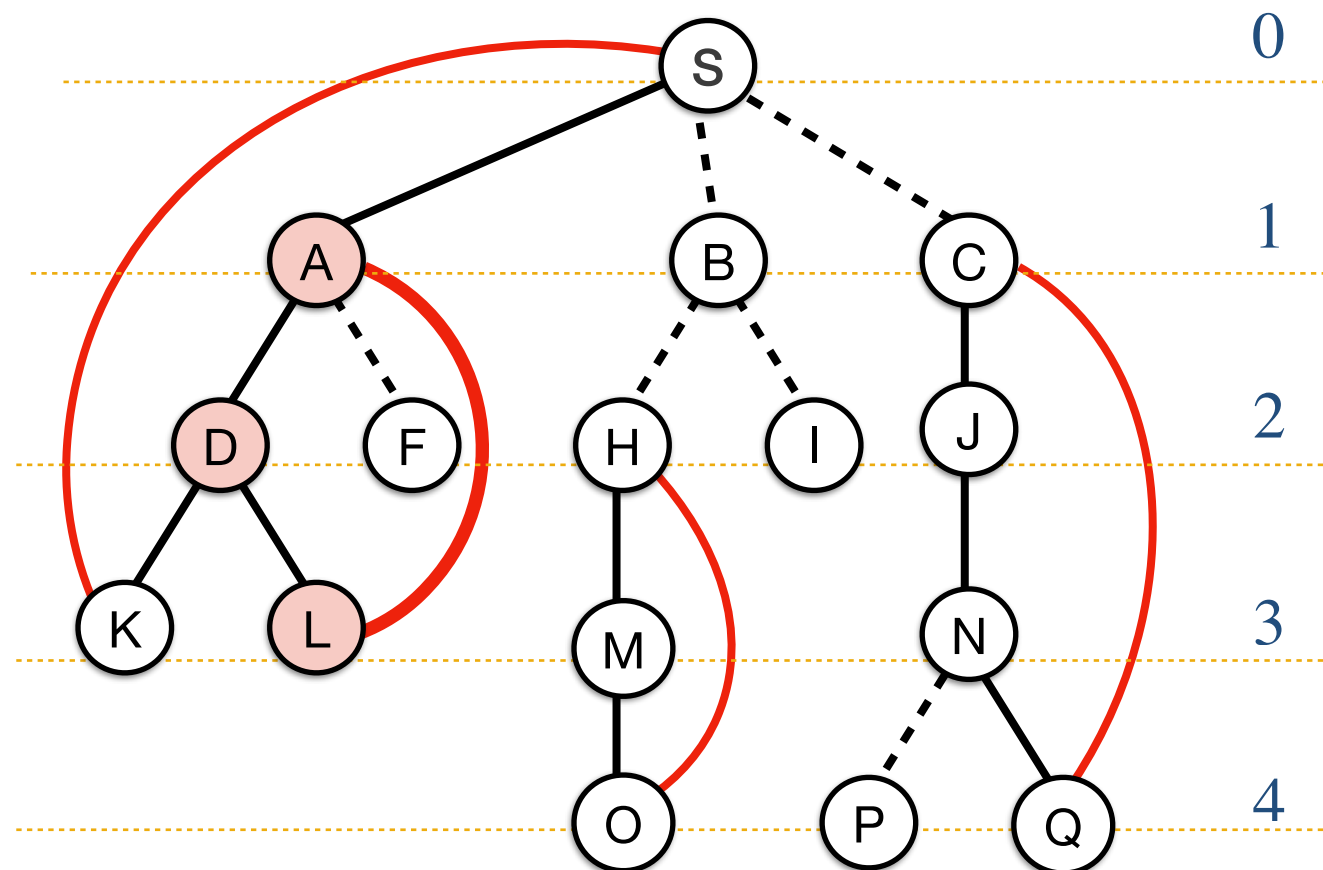
Claim 1: If (x, y) is a non-tree edge of DFS tree T , then all the vertices on cycle obtained by adding (x, y) to T are bi-connected.



Question 2

Let $G = (V, E)$ be an undirected graph. Find a partition $\mathcal{P} = (V_1, \dots, V_k)$ of V in $O(m + n)$ time so that any pair $(x, y) \in V \times V$ is bi-connected iff they lie in same partition.

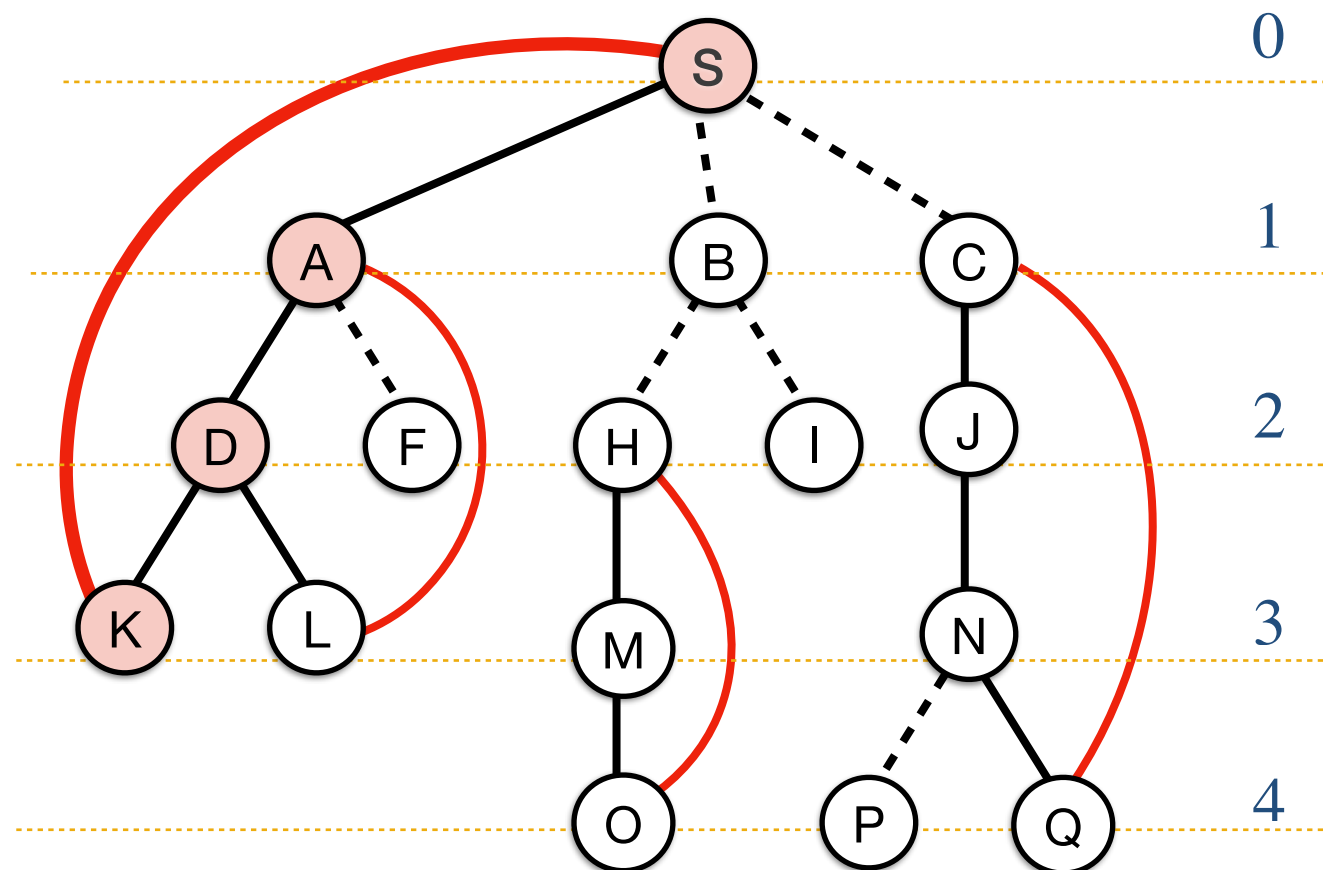
Claim 1: If (x, y) is a non-tree edge of DFS tree T , then all the vertices on cycle obtained by adding (x, y) to T are bi-connected.



Question 2

Let $G = (V, E)$ be an undirected graph. Find a partition $\mathcal{P} = (V_1, \dots, V_k)$ of V in $O(m + n)$ time so that any pair $(x, y) \in V \times V$ is bi-connected iff they lie in same partition.

Claim 1: If (x, y) is a non-tree edge of DFS tree T , then all the vertices on cycle obtained by adding (x, y) to T are bi-connected.



Question 2

Let $G = (V, E)$ be an undirected graph. Find a partition $\mathcal{P} = (V_1, \dots, V_k)$ of V in $O(m + n)$ time so that any pair $(x, y) \in V \times V$ is **bi-connected** iff they lie in same partition.

Claim 2: If x, y are **bi-connected** and y, z are **bi-connected**, then so are vertices x, z .

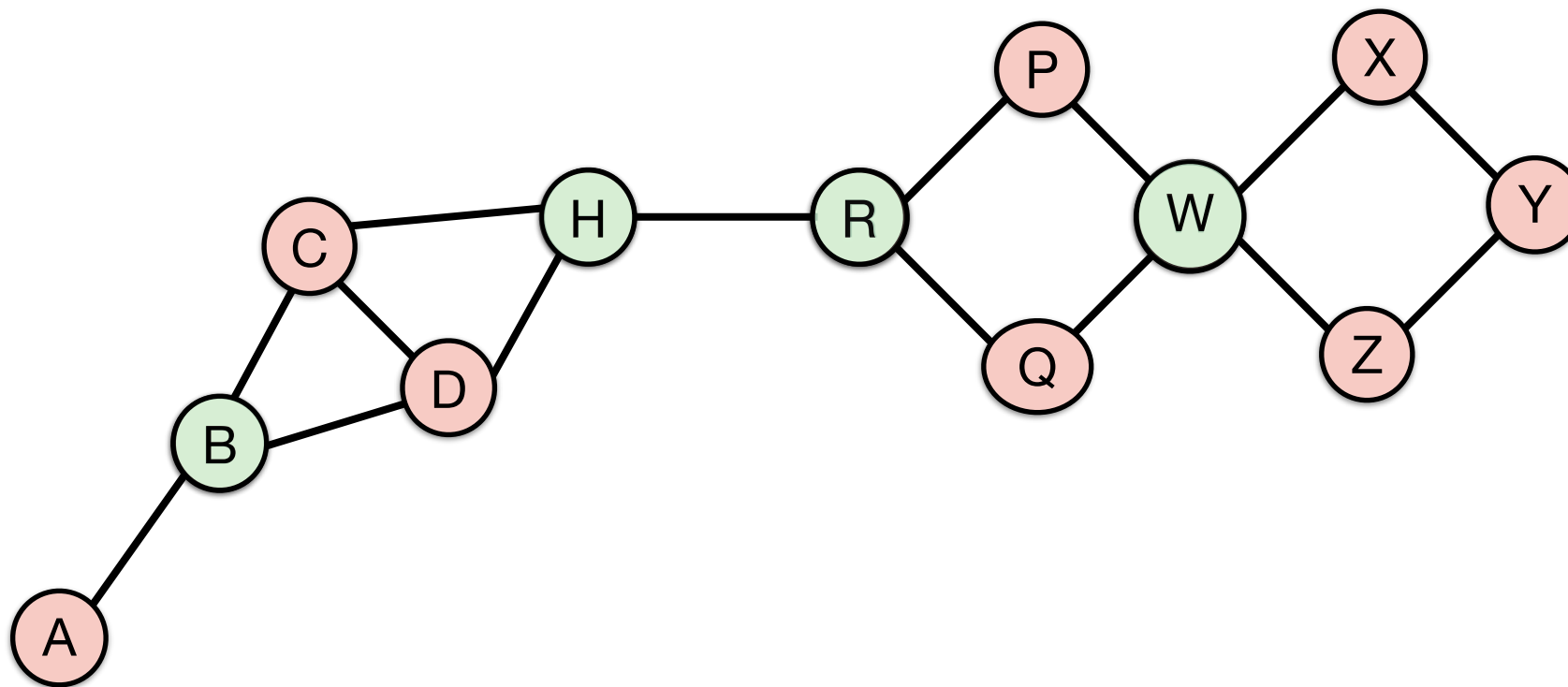
Proof: Homework

Question 3

Let $G = (V, E)$ be an undirected graph. Find cut vertices of G in $O(m + n)$ time using DFS traversal.

Definition:

A vertex x is said to be a **cut vertex** if the number of **connected components of $G - \{x\}$** is larger than the number of **connected components of G** .



Question 3

Let $G = (V, E)$ be an undirected graph. Find cut vertices of G in $O(m + n)$ time using DFS traversal.

Observation 1:

A non-leaf / non-root node x is a **cut vertex** if it has at least one child node, say y , such that

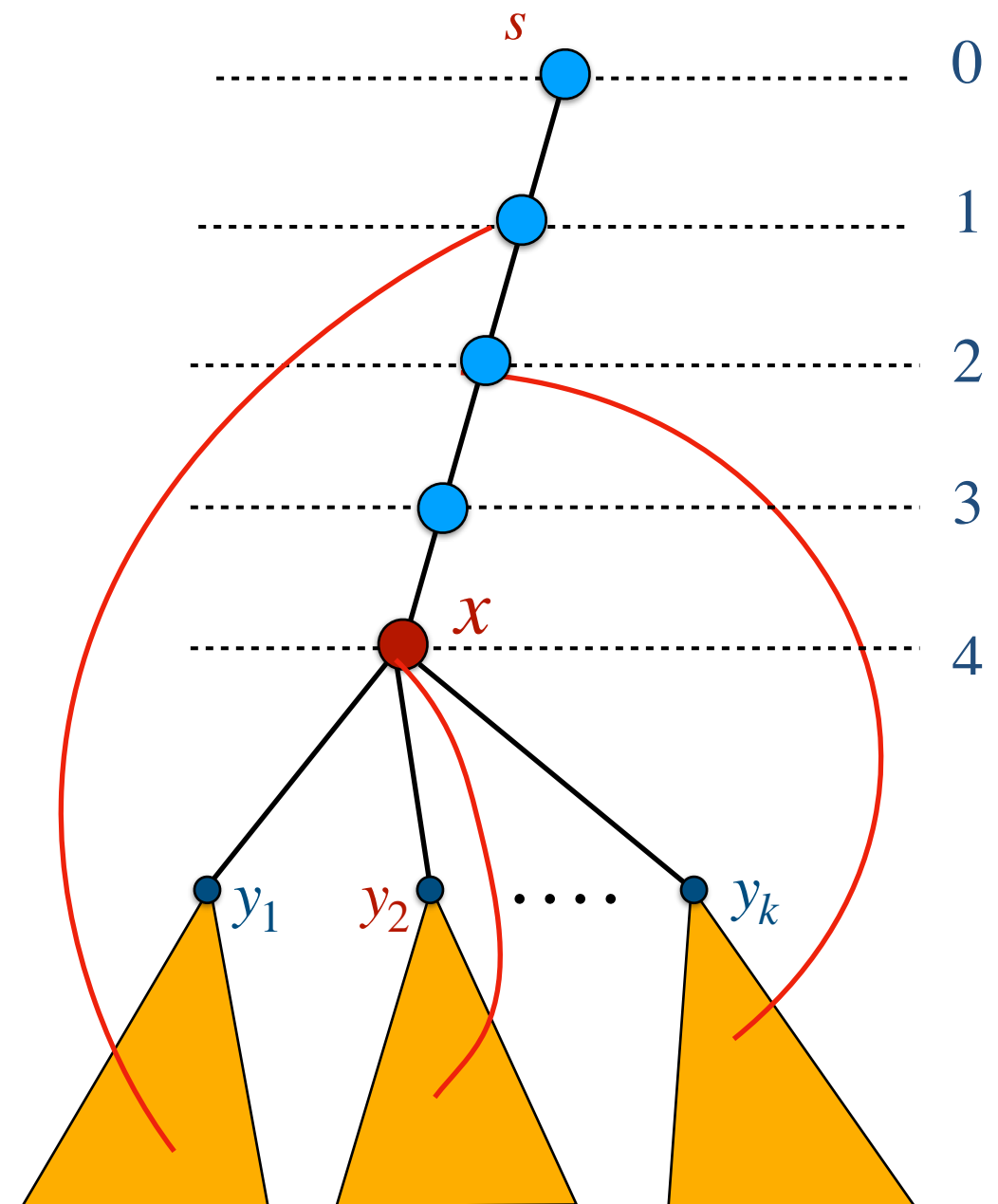
$$HP(y) \geq Level(x)$$

Observation 2:

A leaf node x can never be a **cut vertex**.

Observation 3:

A root node x is a **cut vertex** iff it has two or more children.



Question 4

Let $G = (V, E)$ be a directed acyclic graph (DAG). Provide a pseudocode for computing topological ordering of graph G in $O(m + n)$ time.

Step 1: Compute Finish-time of vertices of G .

Algorithm:

1. counter = 1
2. For each $v \in V$:
 $visited(v) = false$
3. For each $v \in V$:
 If ($visited(v) = false$)
 Invoke DFS(v)

DFS(x):

1. Set $visited(x) = true$
2. For each $y \in N(x)$:
 If ($visited(y) = false$):
 Invoke DFS(y)
3. $FNT(x) = counter++$

Step 2: Return the vertices in the decreasing order of their finish time.

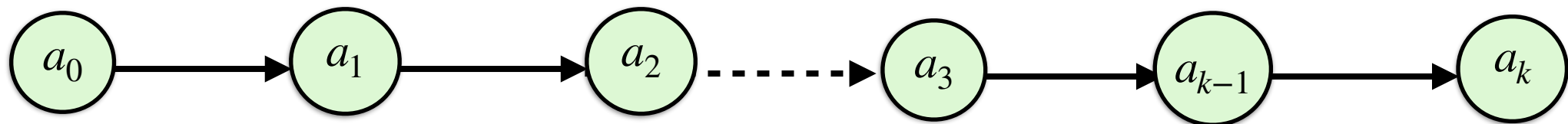
Question 5

Let $G = (V, E)$ be a DAG, and y, z be any two vertices such that there is a path from y to z .
Prove that $FINISH - TIME(z) < FINISH - TIME(y)$ for each DFS traversal in G .

Claim 1: For any directed edge (a, b) in graph G we have:

$$FINISH - TIME(b) < FINISH - TIME(a)$$

Claim 2: For any directed path $(a_0, a_1, \dots, a_k)$ we will have:



$$FINISH - TIME(a_k) < FINISH - TIME(a_{k-1}) < \dots < FINISH - TIME(a_0)$$