



Bridges and Articulation Points

Bui Trong Duc

Hanoi University of Science and Technology

Problem

➤ Description

- Given an undirected graph containing N vertices and M edges

?

Find all the articulation points and the bridges in the graph

➤ Input

- The first line consists of two space-separated integers denoting N and M
- M lines follow, each containing two space-separated integers X and Y denoting there is an edge between X and Y

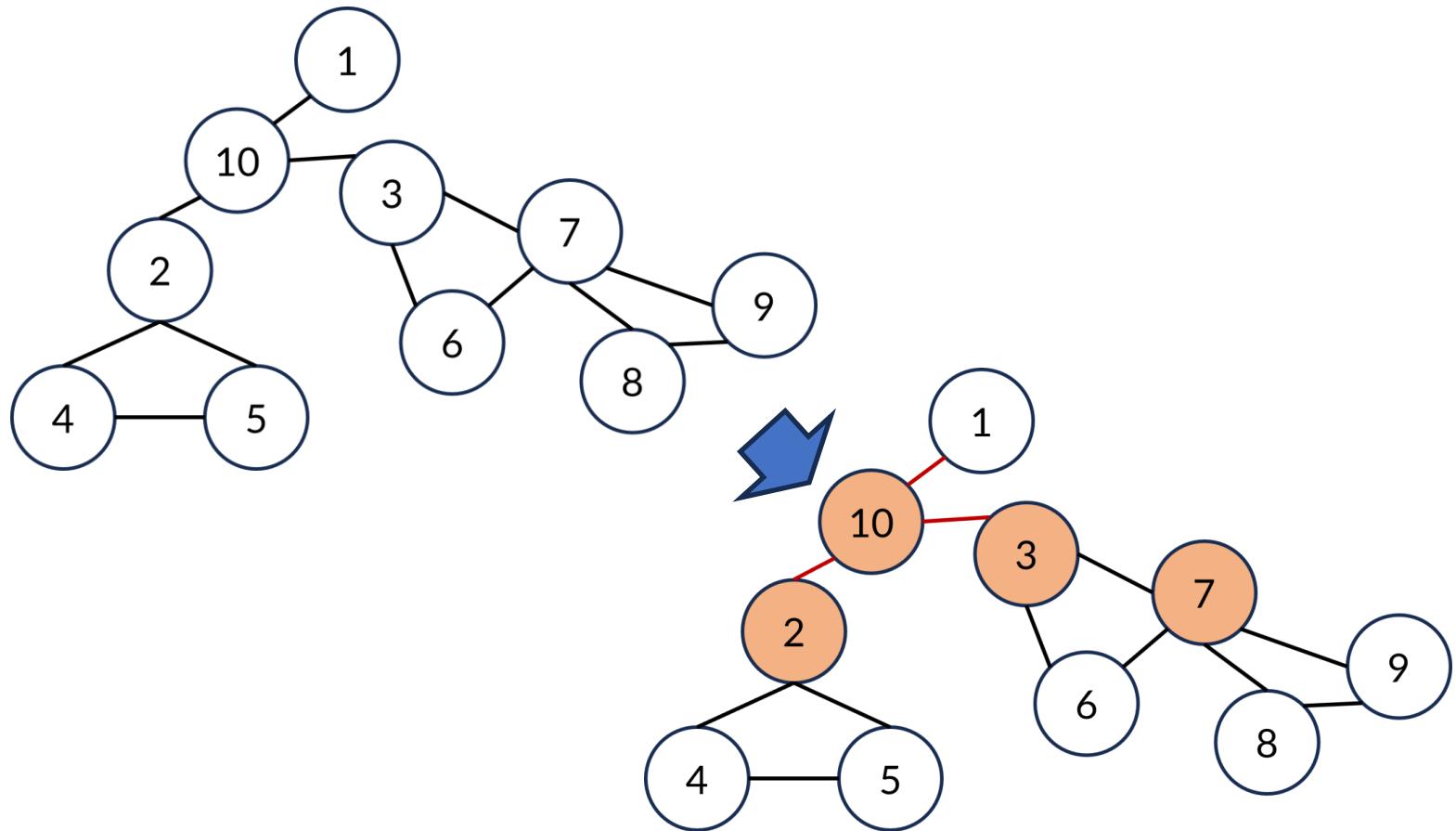
➤ Output

- One line consists of two integers denoting the number of articulation points and the number of bridges

Problem

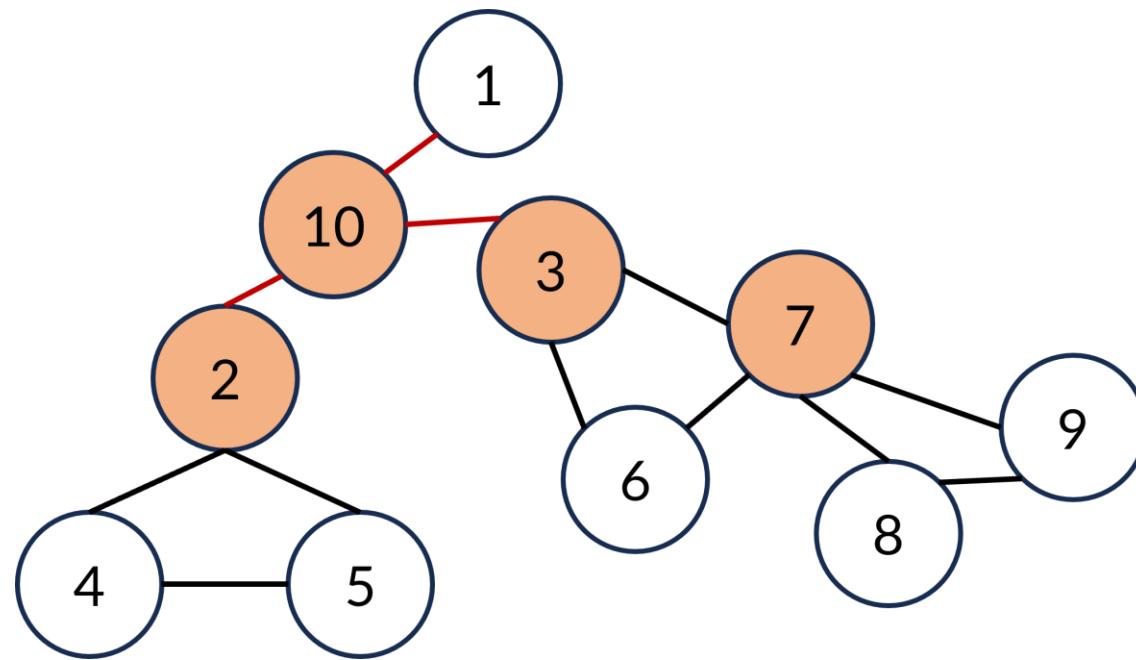
Example

stdin	stdout
10 12	4 3
1 10	
10 2	
10 3	
2 4	
4 5	
5 2	
3 6	
6 7	
7 3	
7 8	
8 9	
9 7	



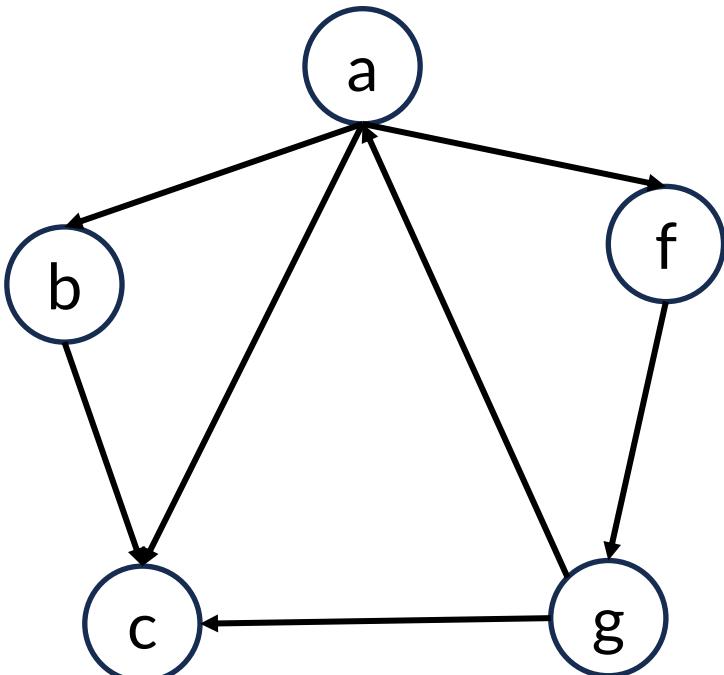
Problem

- Remove articulation points/bridges -> the number of connected components in the graph increase



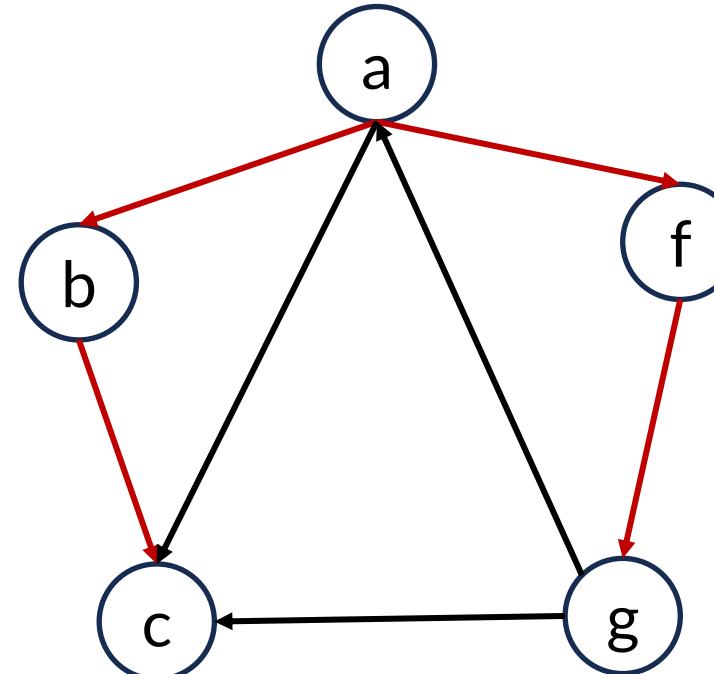
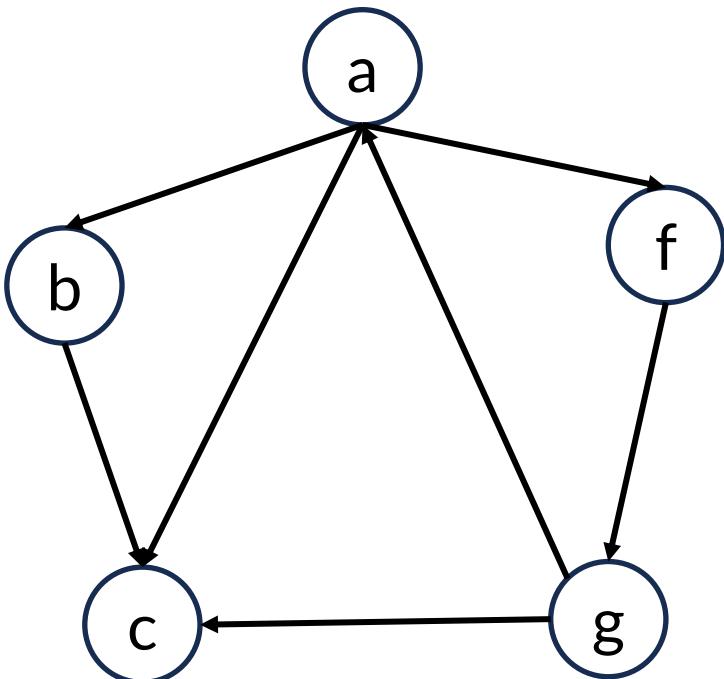
Cây DFS

- Thứ tự duyệt trong DFS theo cơ chế vào sau ra trước (LIFO – Last In First Out), và bắt đầu từ đỉnh xuất phát u nào đó. Đệ quy hoặc Ngăn xếp để hỗ trợ duyệt
- Độ phức tạp tính toán: $O(|V|+|E|)$, trong đó V là tập đỉnh và E là tập cạnh của đồ thị G , vì mỗi đỉnh và mỗi cạnh của G được thăm một lần.



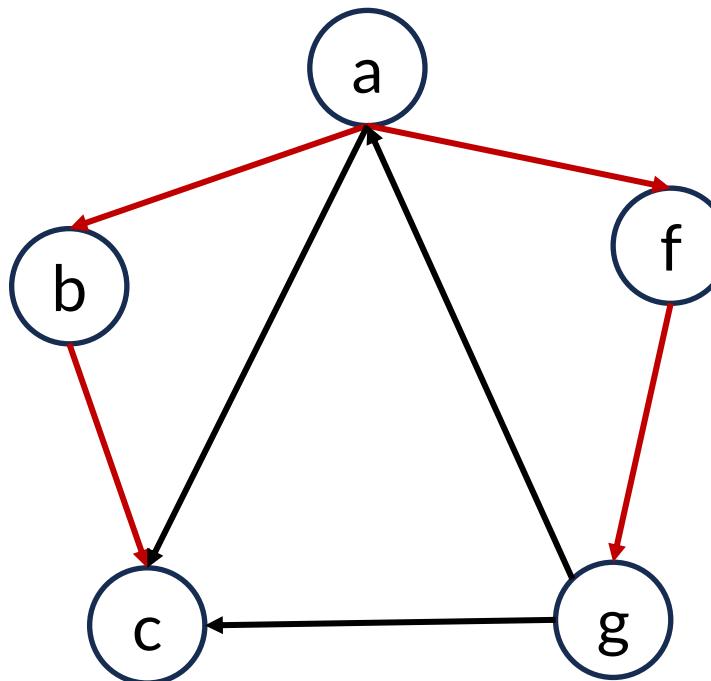
Cây DFS

- Thứ tự duyệt trong DFS theo cơ chế vào sau ra trước (LIFO – Last In First Out), và bắt đầu từ đỉnh xuất phát u nào đó. Đệ quy hoặc Ngăn xếp để hỗ trợ duyệt
- Độ phức tạp tính toán: $O(|V|+|E|)$, trong đó V là tập đỉnh và E là tập cạnh của đồ thị G , vì mỗi đỉnh và mỗi cạnh của G được thăm một lần.

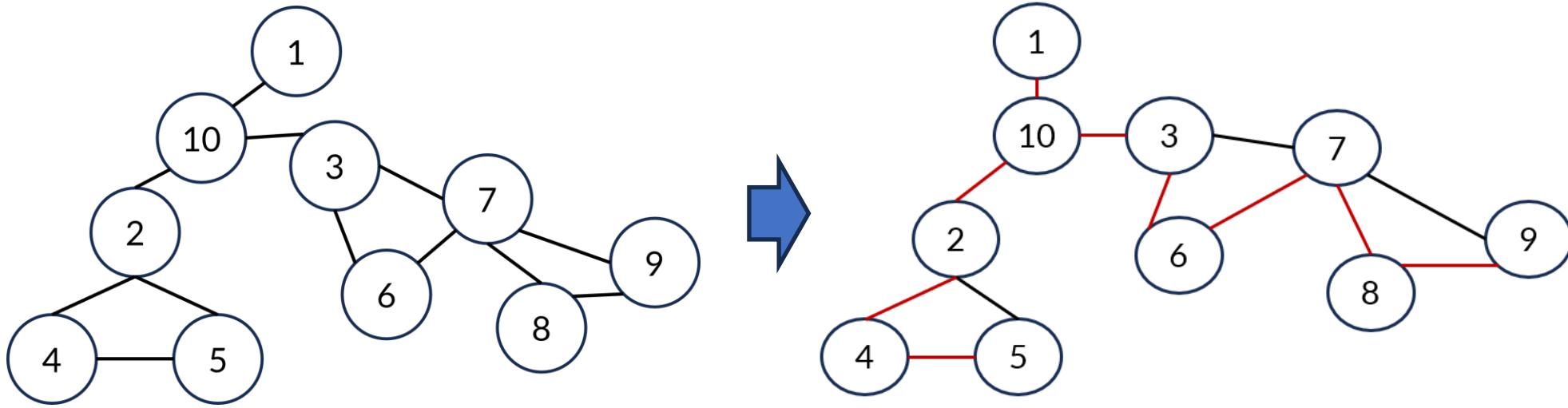


Cây DFS

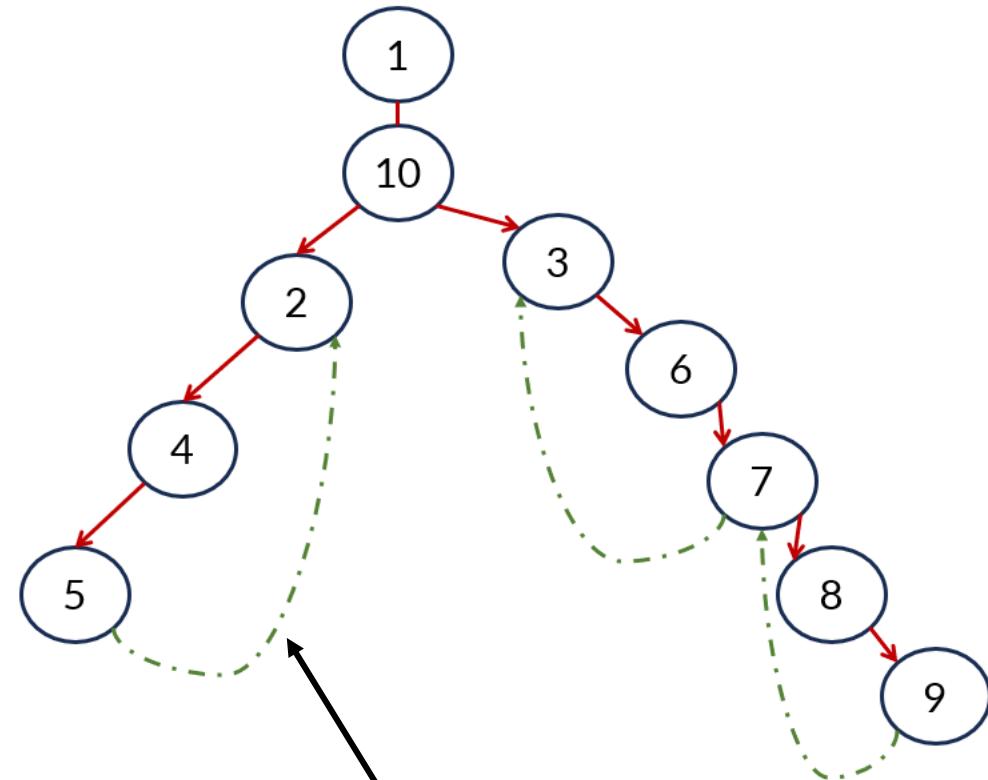
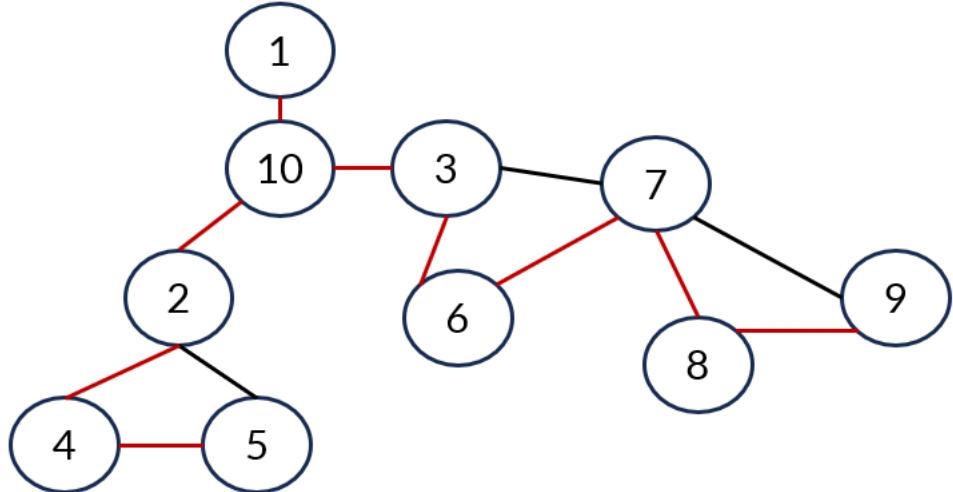
- Cạnh cây (tree edge): từ một đỉnh thăm được đỉnh mới thông qua cạnh
- Cạnh xuôi (forward edge): cạnh đi từ tổ tiên đến con cháu, ví dụ (a,c)
- Cạnh ngược (back edge): cạnh đi từ con cháu đến tổ tiên, ví dụ (g,a)



Cây DFS



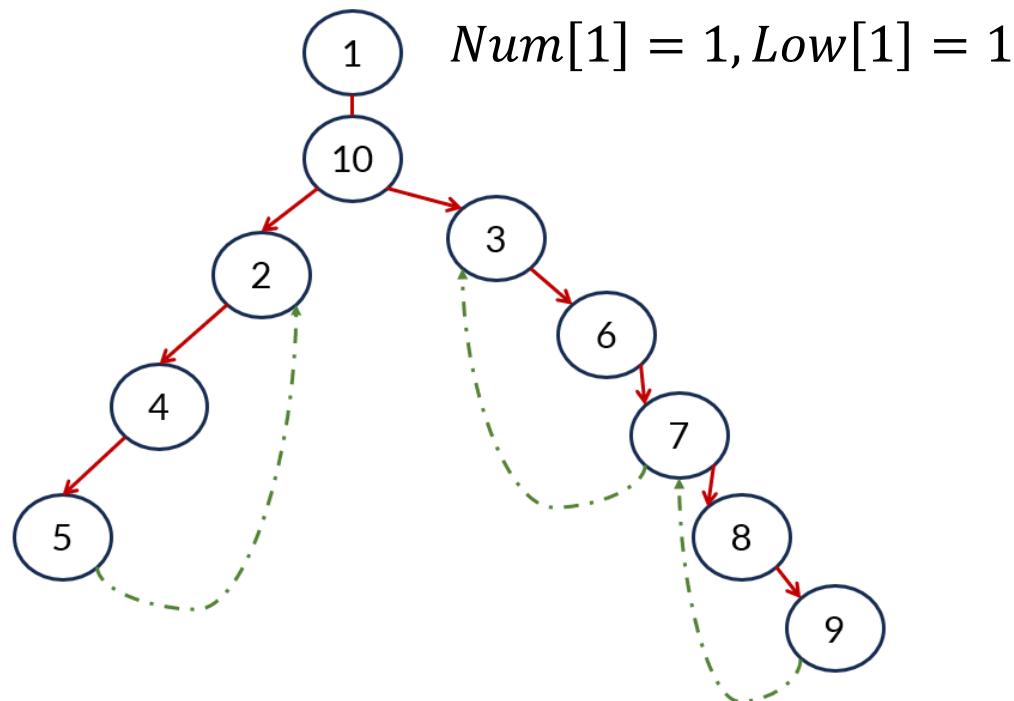
Cây DFS



Cạnh ngược

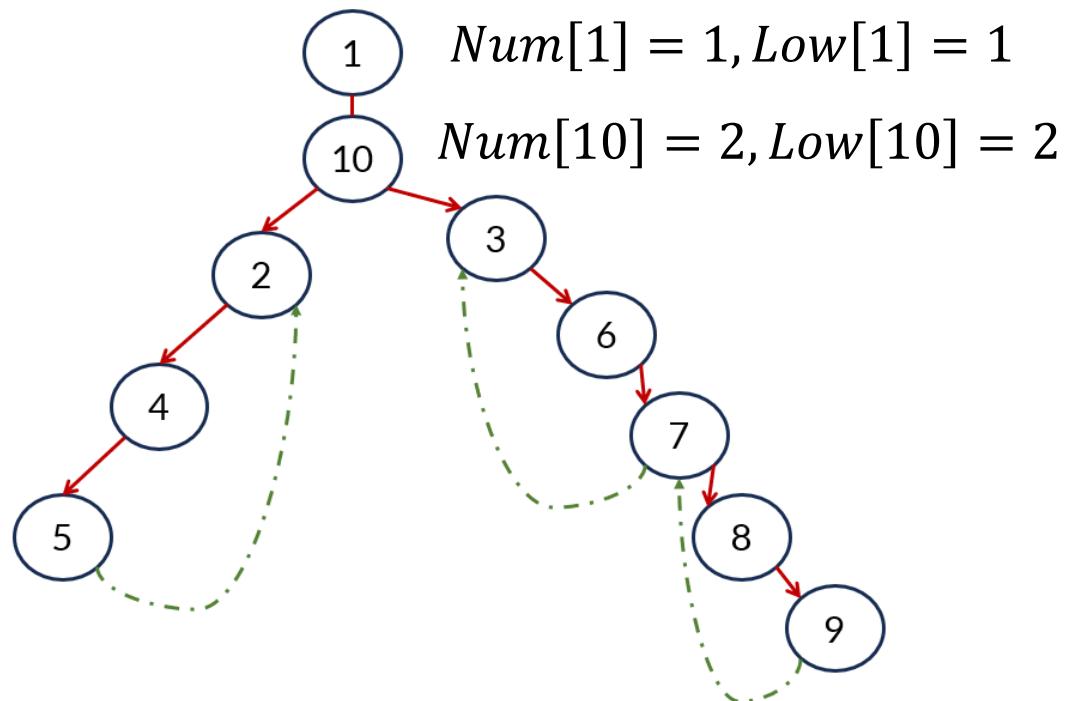
Cây DFS

- Mỗi đỉnh của cây: u
 - $Num[u]$: thứ tự tham của đỉnh u trong DFS
 - $Low[u]$: thứ tự nhỏ nhất mà u hoặc con cháu của u có thể qua trở lại được



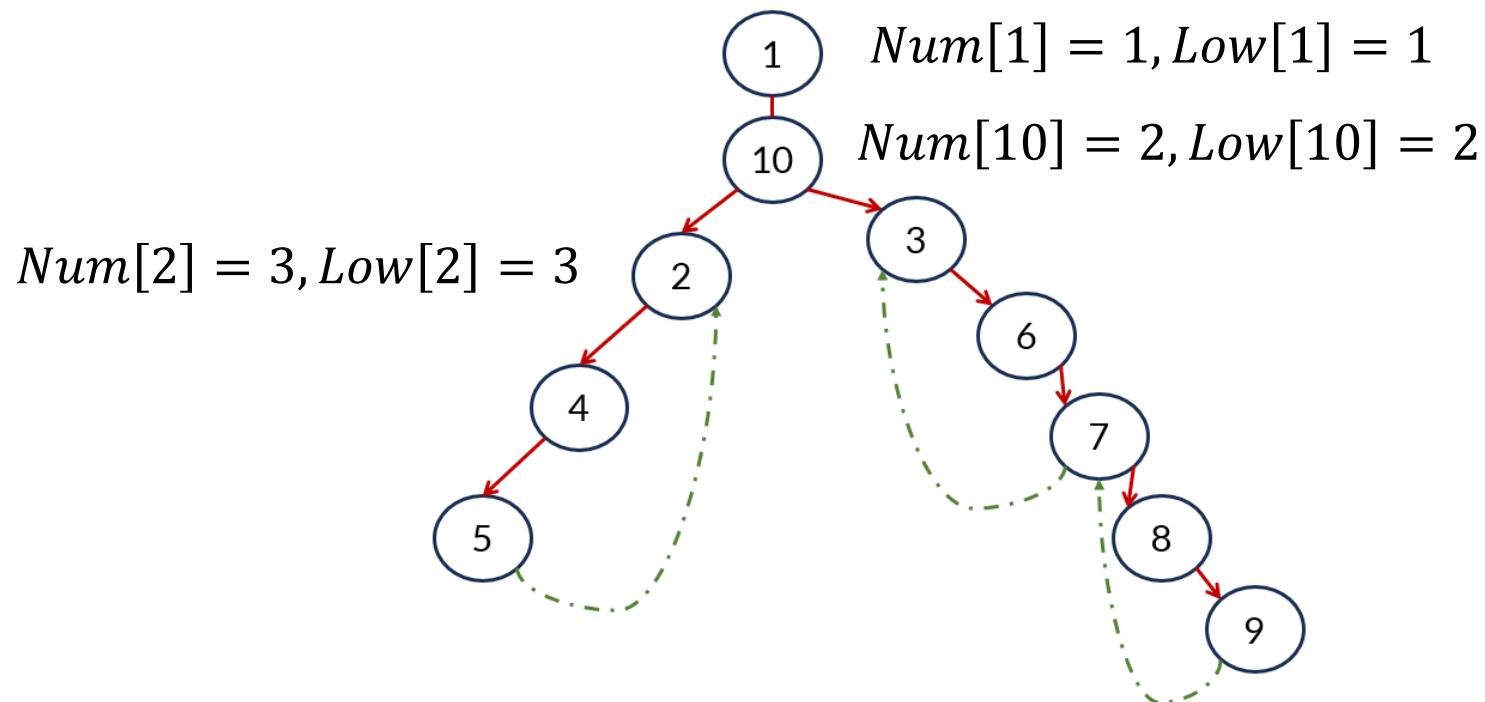
Cây DFS

- Mỗi đỉnh của cây: u
 - $Num[u]$: thứ tự tham của đỉnh u trong DFS
 - $Low[u]$: thứ tự nhỏ nhất mà u hoặc con cháu của u có thể qua trở lại được



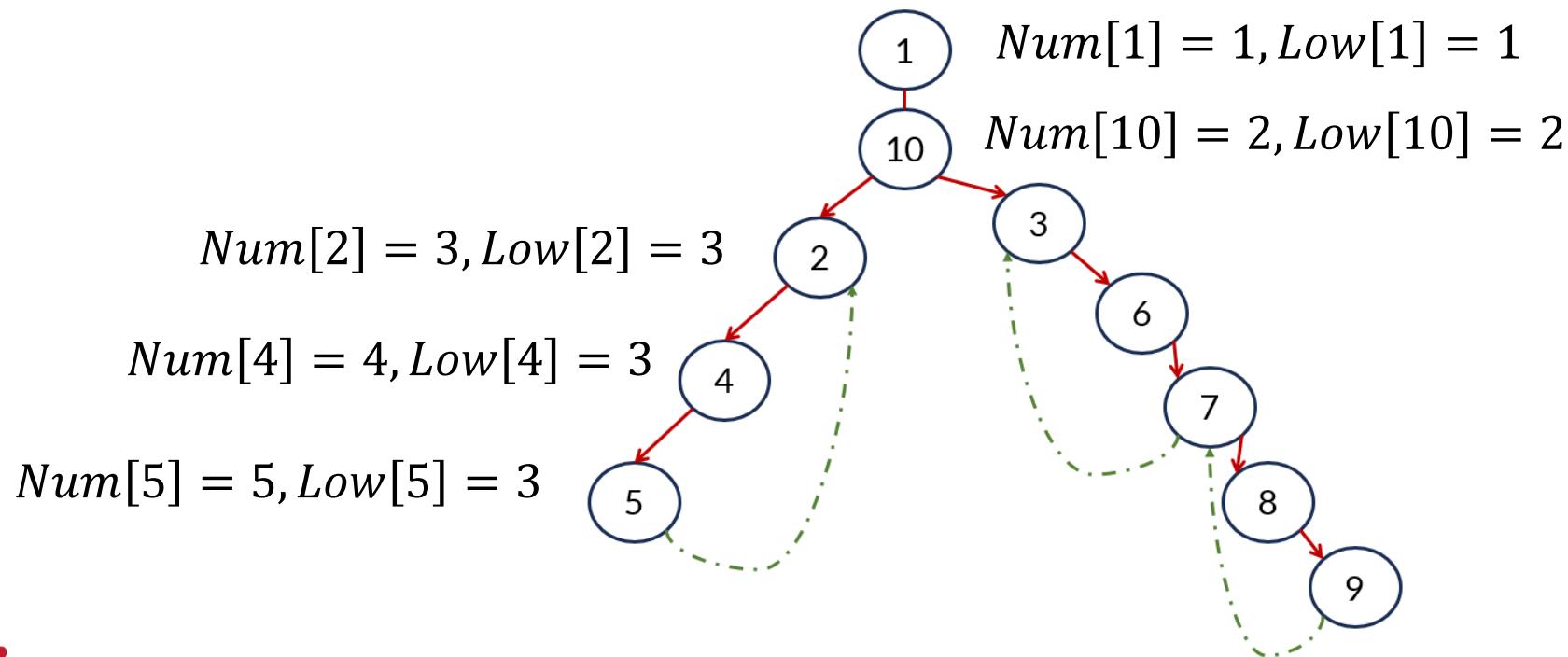
Cây DFS

- Mỗi đỉnh của cây: u
 - $Num[u]$: thứ tự tham của đỉnh u trong DFS
 - $Low[u]$: thứ tự nhỏ nhất mà u hoặc con cháu của u có thể qua trở lại được



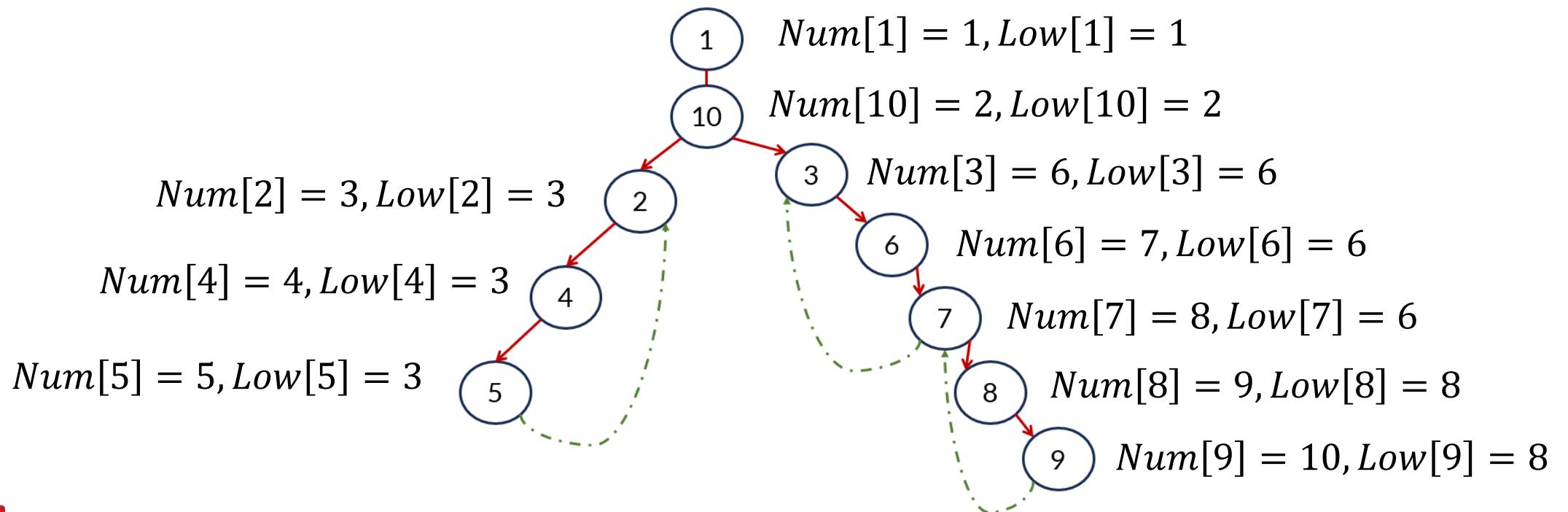
Cây DFS

- Mỗi đỉnh của cây: u
 - $Num[u]$: thứ tự tham của đỉnh u trong DFS
 - $Low[u]$: thứ tự nhỏ nhất mà u hoặc con cháu của u có thể qua trở lại được



Cây DFS

- Mỗi đỉnh của cây: u
 - $Num[u]$: thứ tự tham của đỉnh u trong DFS
 - $Low[u]$: thứ tự nhỏ nhất mà u hoặc con cháu của u có thể qua trở lại được



Cây DFS

- Mỗi đỉnh của cây: u
 - $Num[u]$: thứ tự tham của đỉnh u trong DFS
 - $Low[u]$: thứ tự nhỏ nhất mà u hoặc con cháu của u có thể qua trở lại được
 - $Low[u]$: giá trị nhỏ nhất trong những giá trị sau:
 - $Num[v]$ nếu (v, u) là một cạnh ngược \\\(v đã được thăm
 - $Low[v]$ nếu v là con cháu của u trong cây DFS \\\(v đã được thăm lần đầu
 - $Num[u]$ \\\(không có cạnh ngược

Phương pháp

- Sử dụng DFS xây dựng Num và Low
- Một cạnh xuôi (u, v) là cầu khi và chỉ khi $Low[v] > Num[u]$
- Đỉnh u được gọi là khớp nếu
 - Đỉnh u là gốc và có ít nhất 2 cây con
 - Đỉnh u không là gốc và $Low[v] \geq Num[u]$ (với v là một con trực tiếp bất kỳ của u trong cây DFS)

THANK YOU