

# Decomposition of Graphs: Previsit and Postvisit Orders

Daniel Kane

Department of Computer Science and Engineering  
University of California, San Diego

Graph Algorithms  
Data Structures and Algorithms

# Learning Objectives

- Compute the preorder and postorder numbers for a DFS.
- Understand why these numbers might be important.

# Outline

1 Definition

2 Properties

# Need to Record Data

- Plain DFS just marks all vertices as visited.
- Need to keep track of other data to be useful.
- Augment functions to store additional information.

# Previsit and Postvisit Functions

Explore( $v$ )

visited( $v$ )  $\leftarrow$  true

previsit( $v$ )

for ( $v, w$ )  $\in E$ :

    if not visited( $w$ ):

        explore( $w$ )

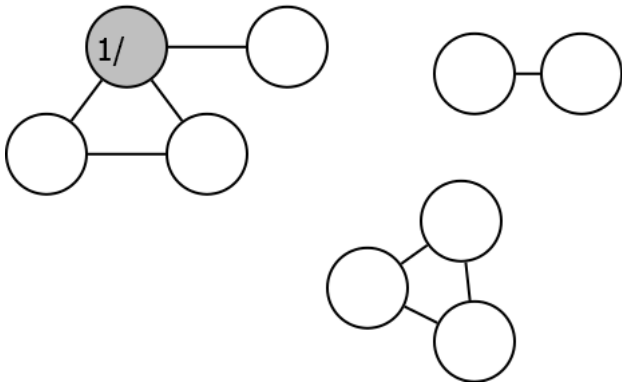
postvisit( $v$ )

# Clock

- Keep track of order of visits.
- Clock ticks at each pre-/post- visit.
- Records previsit and postvisit times for each  $v$ .

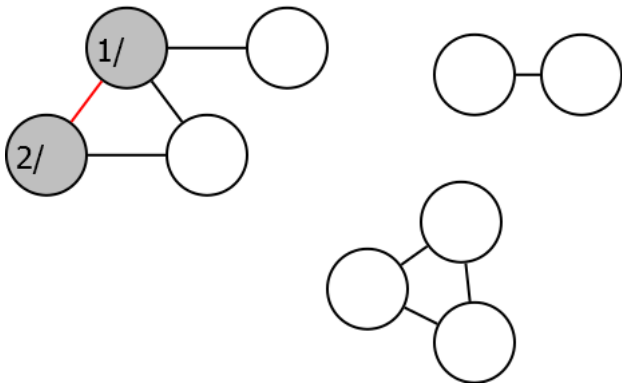
# Example

**Clock: 1**



# Example

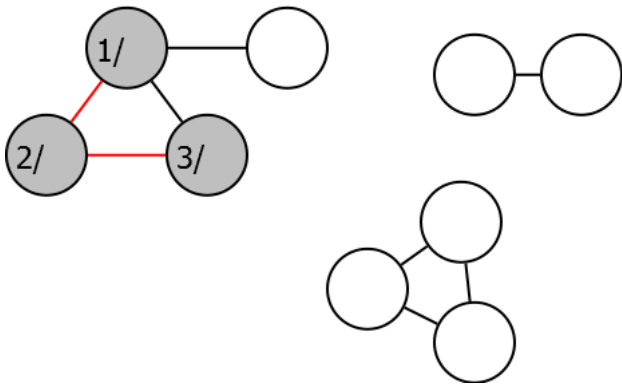
**Clock: 2**





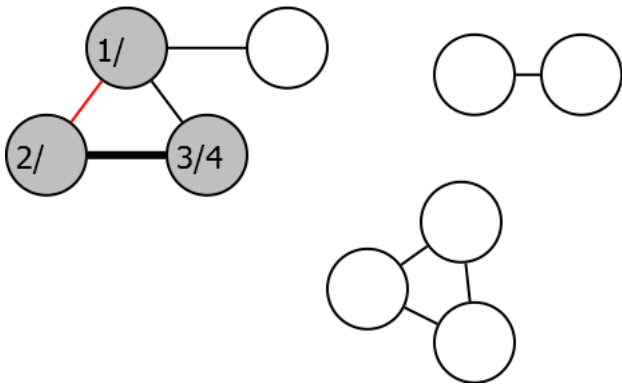
# Example

Clock: 3



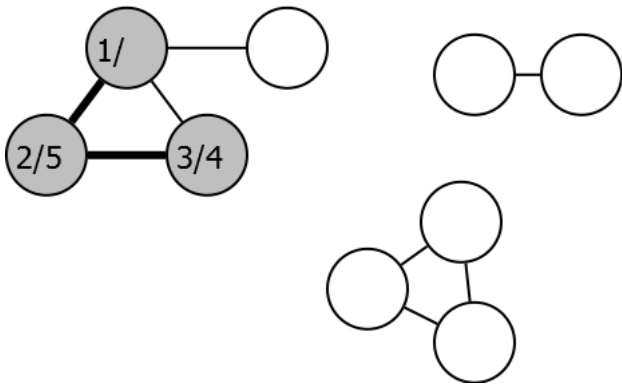
# Example

**Clock: 4**



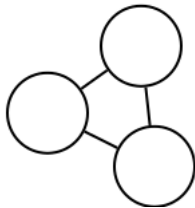
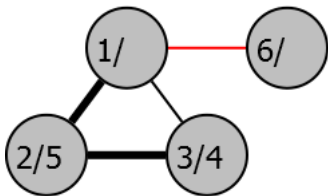
# Example

Clock: 5



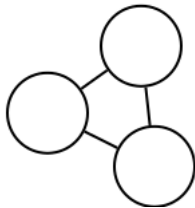
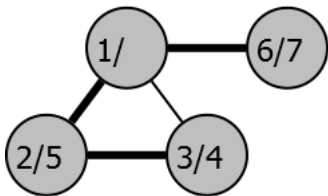
# Example

Clock: 6



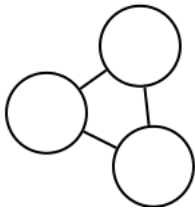
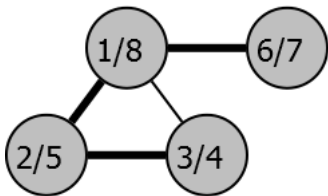
# Example

Clock: 7



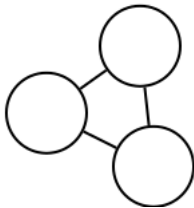
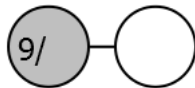
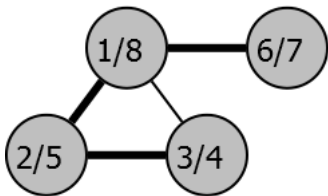
# Example

Clock: 8



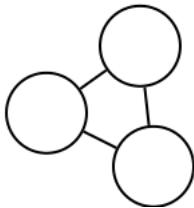
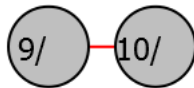
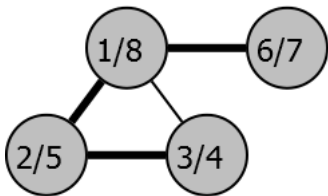
# Example

Clock: 9



# Example

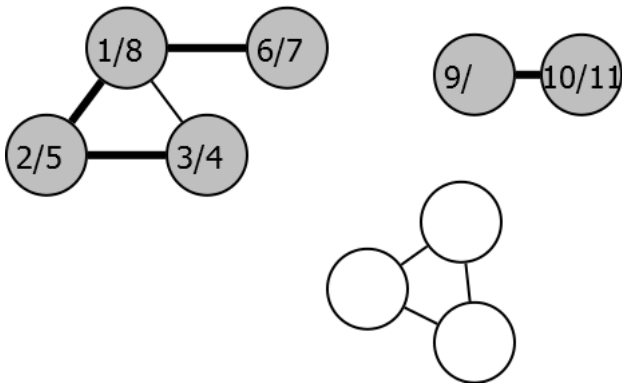
Clock:10





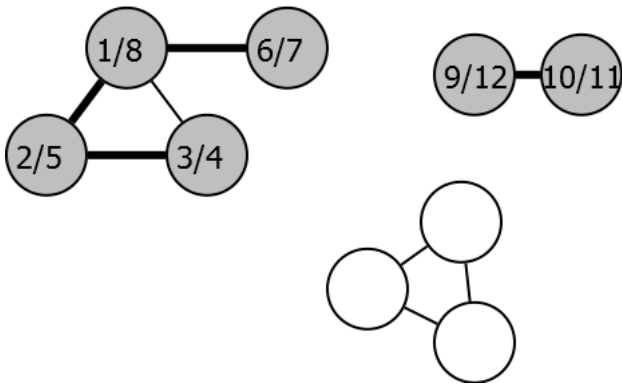
# Example

Clock:11



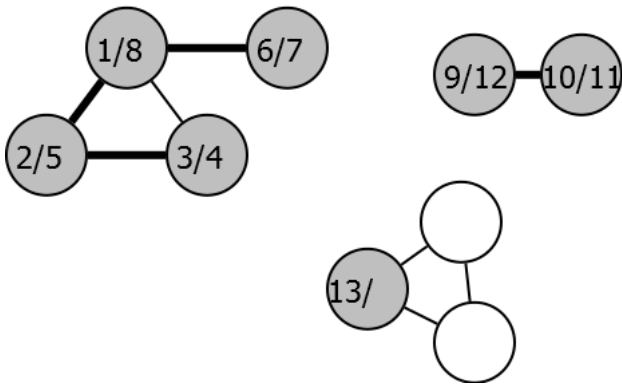
# Example

Clock:12



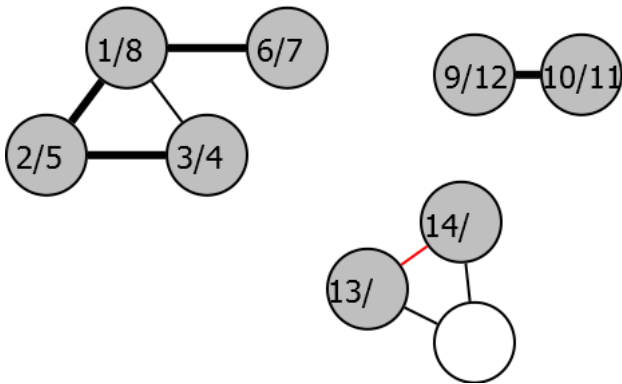
# Example

Clock:13



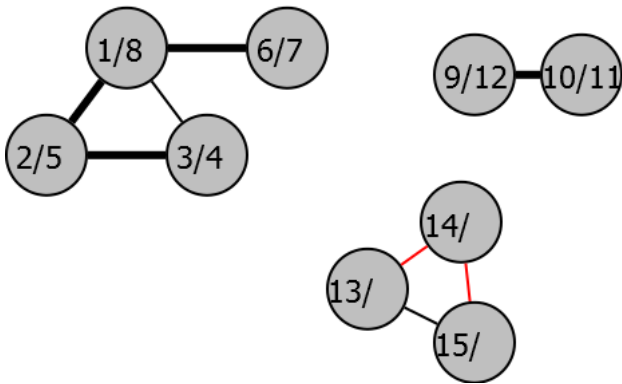
# Example

Clock:14



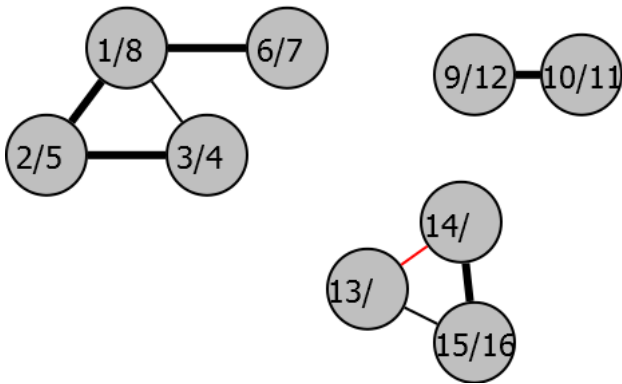
# Example

Clock:15



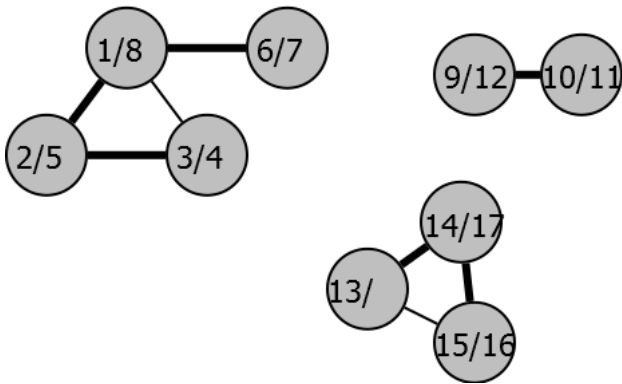
# Example

Clock:16



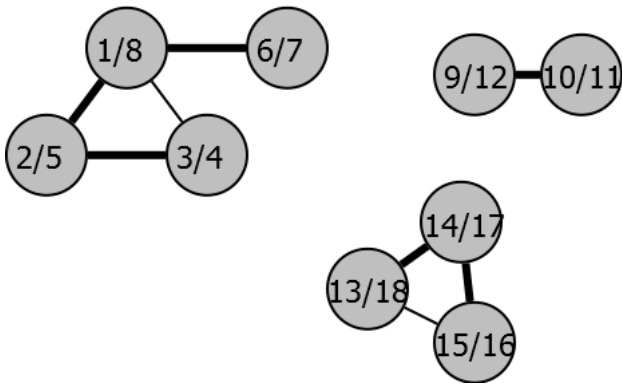
# Example

Clock:17



# Example

Clock:18





# Computing Pre- and Post- Numbers

Initialize clock to 1.

**previsit( $v$ )**

$\text{pre}(v) \leftarrow \text{clock}$

$\text{clock} \leftarrow \text{clock} + 1$

**postvisit( $v$ )**

$\text{post}(v) \leftarrow \text{clock}$

$\text{clock} \leftarrow \text{clock} + 1$

# Outline

1 Definition

2 Properties

# Result

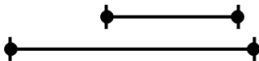
Previsit and Postvisit numbers tell us about the execution of DFS.

## Lemma

For any vertices  $u, v$  the intervals  $[\text{pre}(u), \text{post}(u)]$  and  $[\text{pre}(v), \text{post}(v)]$  are either nested or disjoint.

# Explanation

Nested

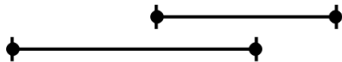


Disjoint



# Explanation

Interleaved (not possible)



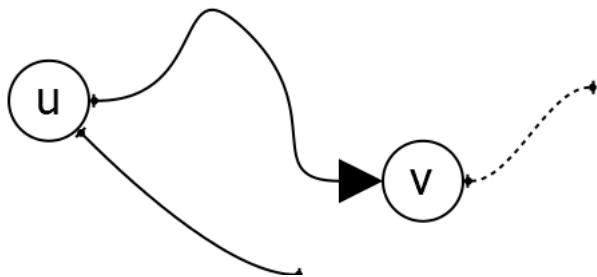
# Proof

Assume that  $u$  visited before  $v$ . Two cases

- Find  $v$  while exploring  $u$  ( $u$  an ancestor of  $v$ )
- Find  $v$  after exploring  $u$  ( $u$  a cousin of  $v$ )

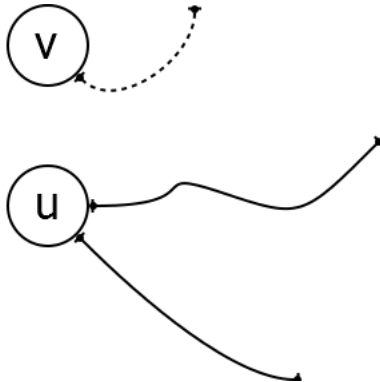
## Case I

If you explore  $v$  while exploring  $u$ , cannot finish exploring  $u$  until done exploring  $v$ .  
Therefore nested.



## Case II

If explore  $v$  after finish exploring  $u$ ,  
 $\text{post}(u) < \text{pre}(v)$ , therefore disjoint.





# Problem

Which of the following tables is not a valid set of pre- and post- orders?

Vert.	Pre	Post
A	1	8
B	9	10
C	3	4
D	2	7
E	5	6

Vert.	Pre	Post
A	1	9
B	8	10
C	2	7
D	3	6
E	4	5

# Solution

Which of the following tables is not a valid set of pre- and post- orders?

Vert.	Pre	Post
A	1	8
B	9	10
C	3	4
D	2	7
E	5	6

Vert.	Pre	Post
A	1	9
B	8	10
C	2	7
D	3	6
E	4	5

# Next Time

Directed graphs.