

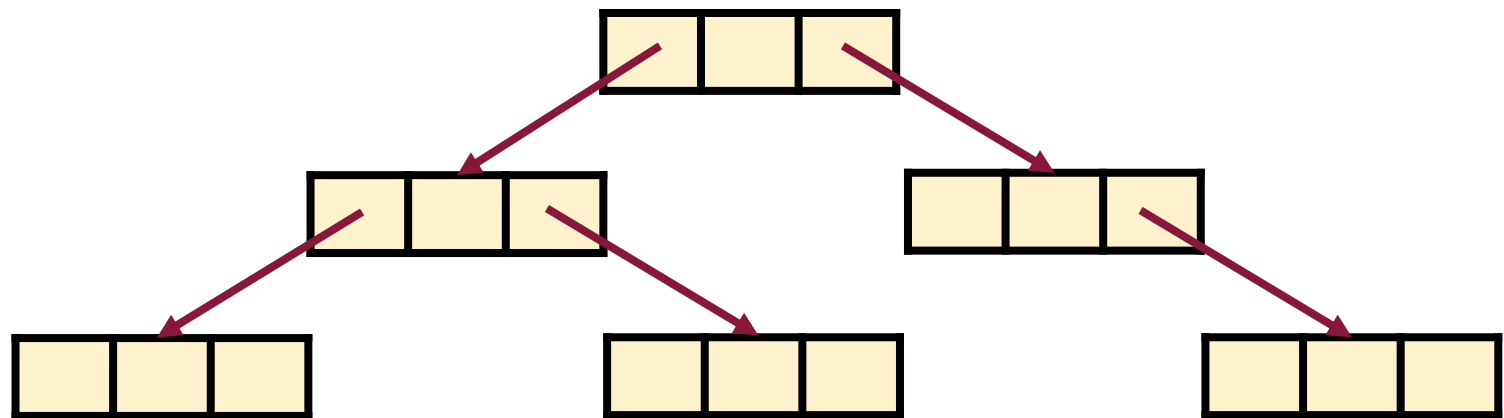
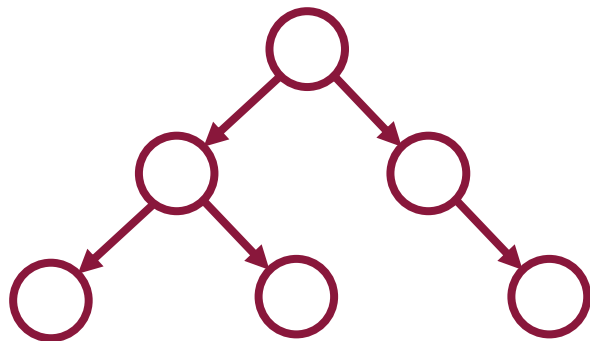


# Binary tree

AIE 311 : Data structure and Algorithm

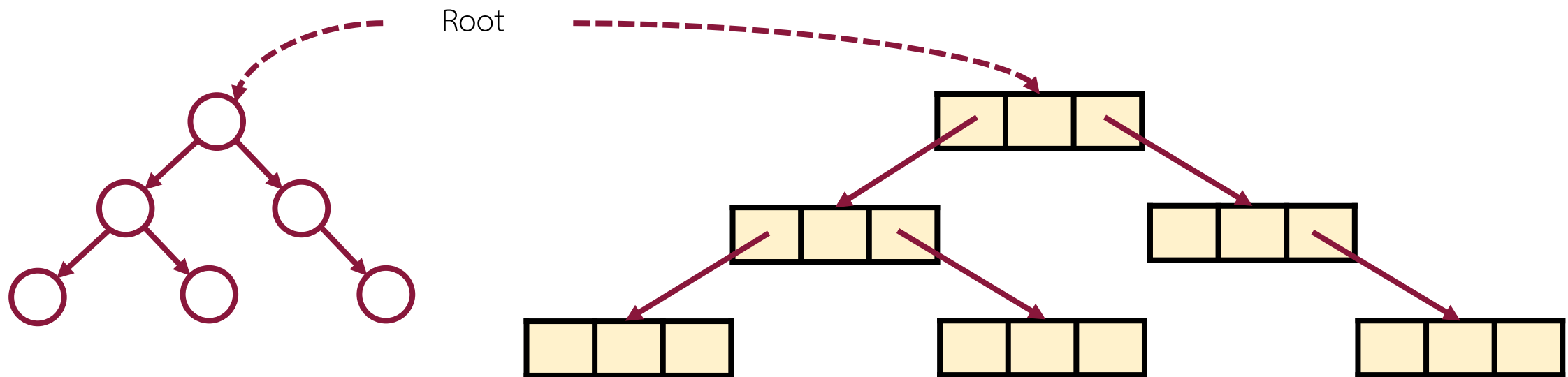


- Structure of binary tree
  - Meaning of “bi” is two. So binary tree is the tree that each node will not contains next value more than two nodes.
  - If observe closely the structure of binary tree will be similar to forward and backward linked list but different purpose.





- Root
  - Root is the header of binary tree and can describe as starting point of binary tree.



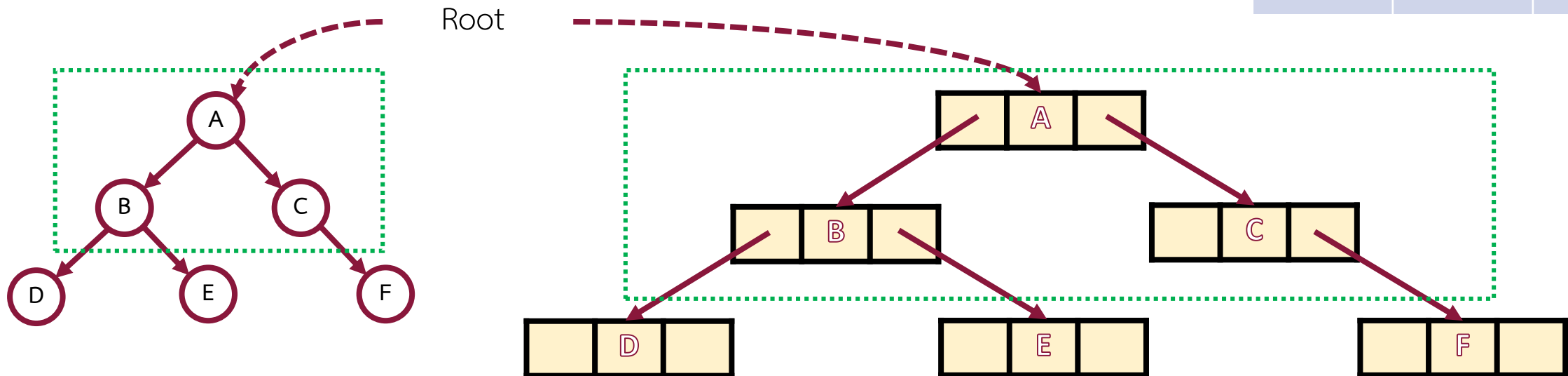
# Indegree and outdegree



- Indegree

- Indegree is where the arrow point at node as input or linked by.
  - i.e, B has 1 indegree..
- Outdegree is where the arrow starting point at node as output or linked
  - to. i.e, A has 2 outdegrees.

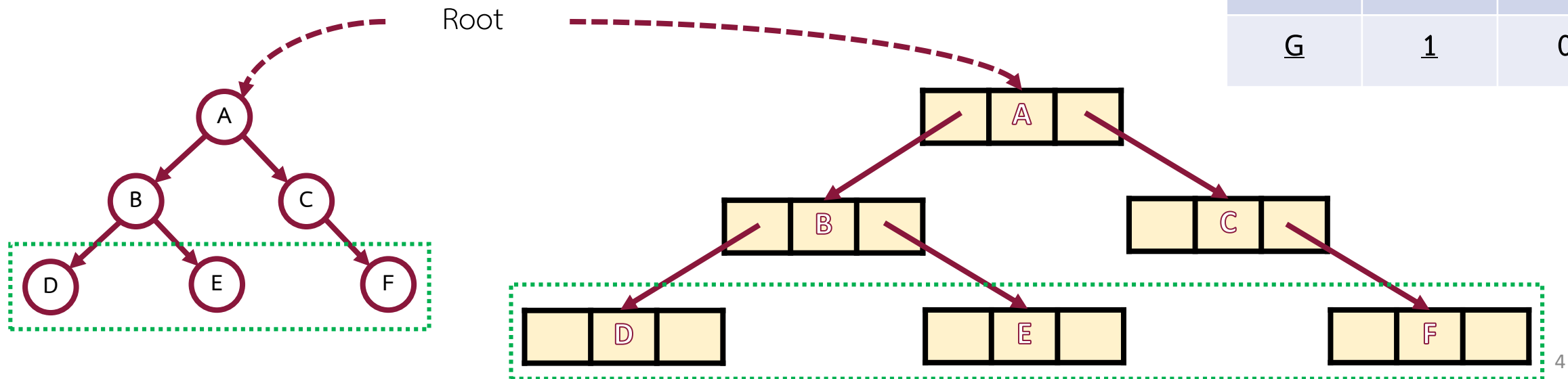
Node	In	Out
A	<u>0</u>	<u>2</u>
B	<u>1</u>	<u>2</u>
C	<u>1</u>	<u>1</u>





- Leaf
  - Leaf is the node does not contain any outdegree.

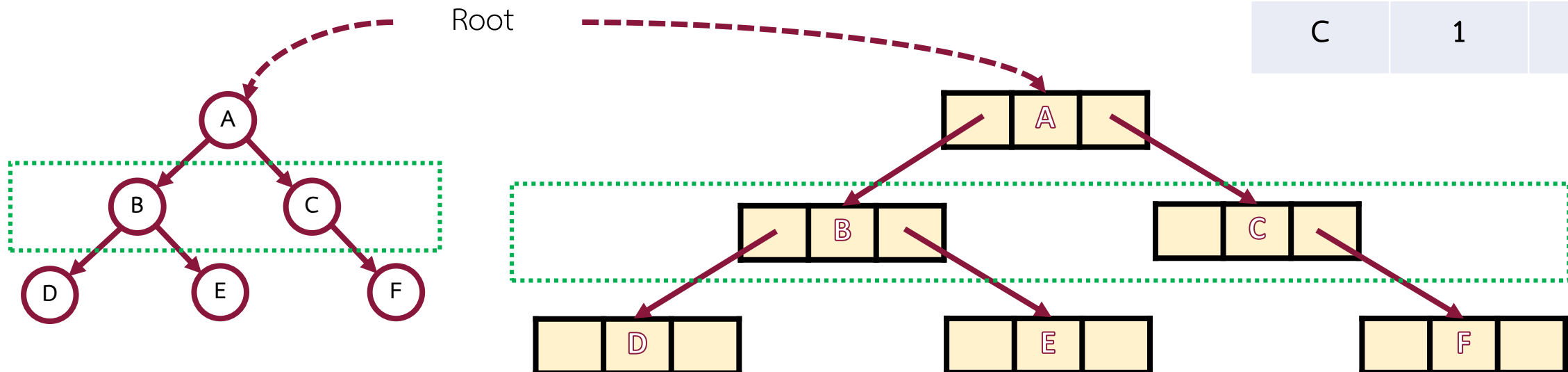
Node	In	Out
<u>D</u>	<u>1</u>	0
<u>E</u>	<u>1</u>	0
<u>F</u>	<u>1</u>	0
<u>G</u>	<u>1</u>	0





- Internal node
  - The internal node is the node that is not root and leaf. This means the node that contains indegree and outdegree at the same time is the internal node.

Node	In	Out
B	1	2
C	1	2



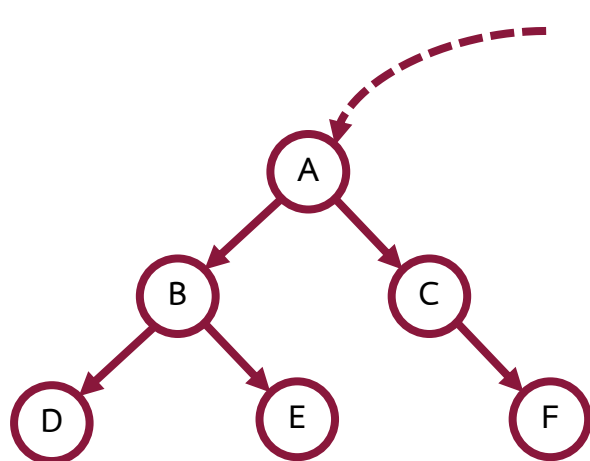


# Parent node and child node

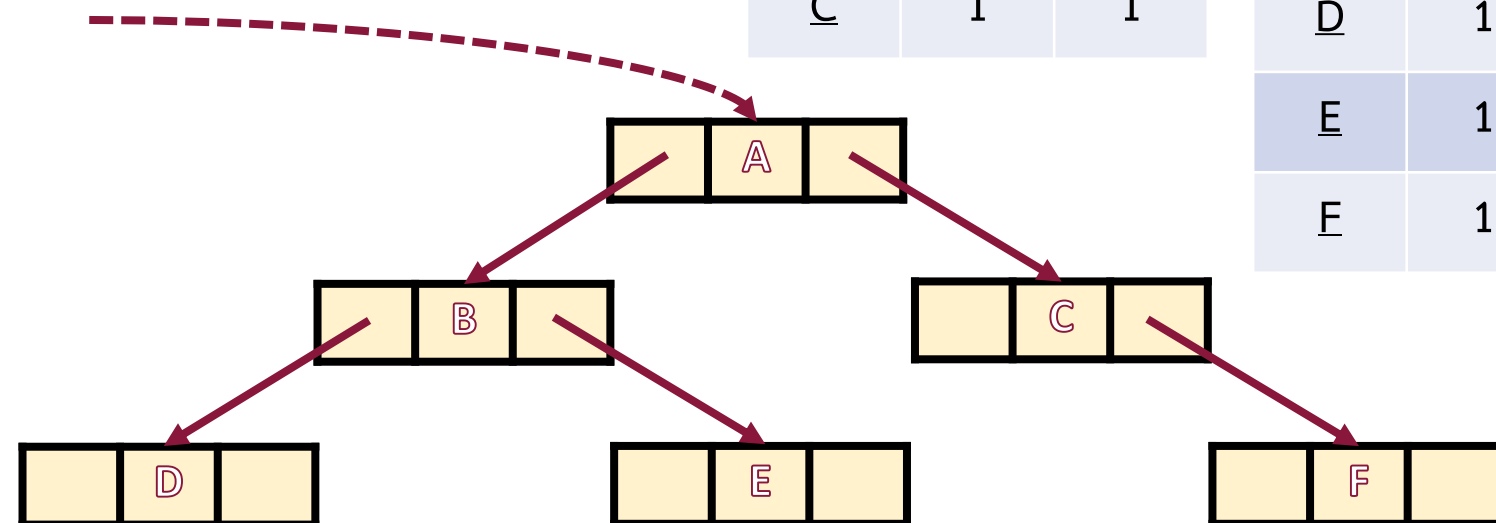
- Parent node
  - The parent node means the node that contains outdegree.
- Child node
  - The child node means the node that contains indegree.

Parent		
Node	In	Out
<u>A</u>	0	2
<u>B</u>	1	2
<u>C</u>	1	1

Child		
Node	In	Out
<u>B</u>	1	2
<u>C</u>	1	2
<u>D</u>	1	0
<u>E</u>	1	0
<u>F</u>	1	0



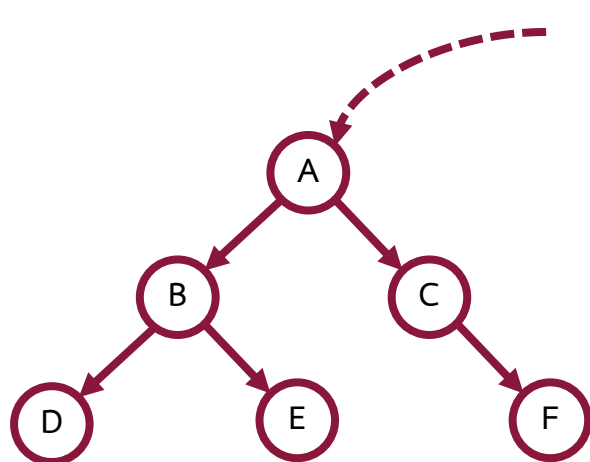
Root



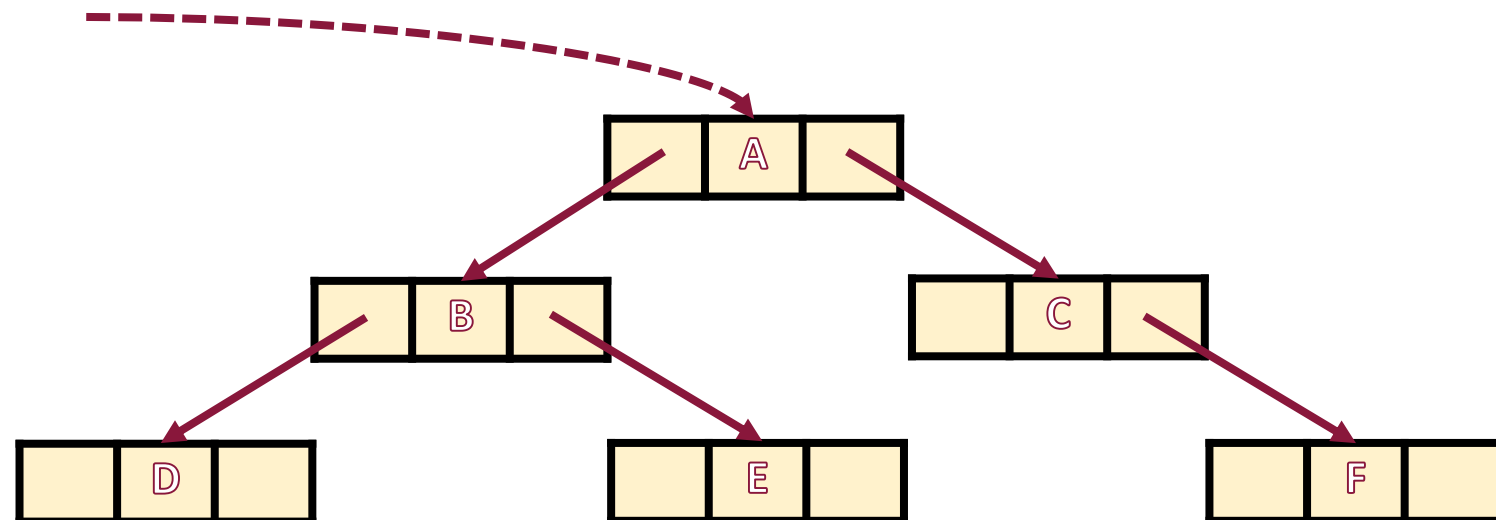


- Sibling
  - Sibling are the nodes those have the same parent.

Parent	Sibling 1	Sibling 2	
A	B	C	{B, C}
B	D	E	{D, E}
C	F		{F}



Root

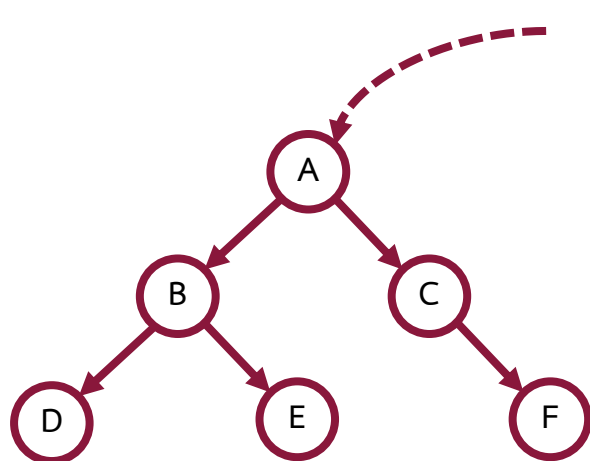




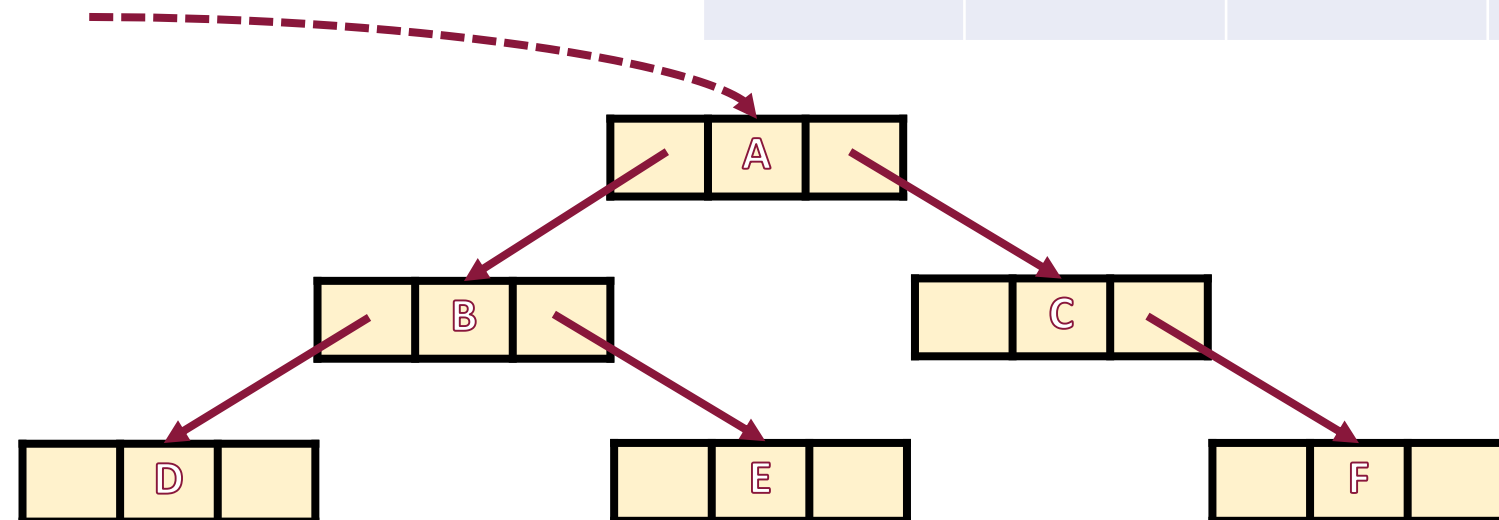
- Path

- The path will show every node from starting point to ending point.
- Nevertheless, every node will contain only 1 path.

Path	Start	End	Output
A – F	A	F	A -> C -> F
A – D	A	D	A -> B -> D



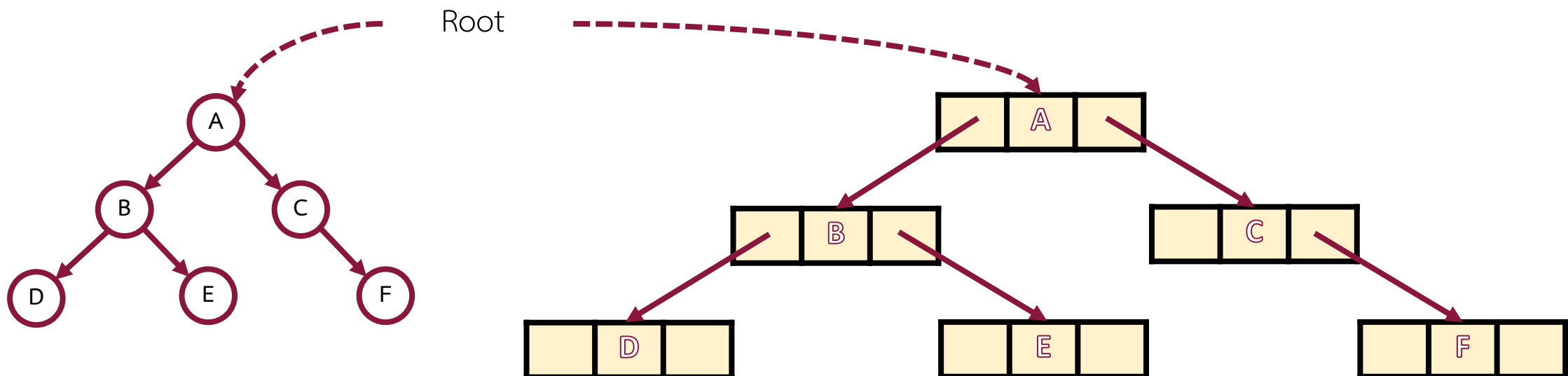
Root





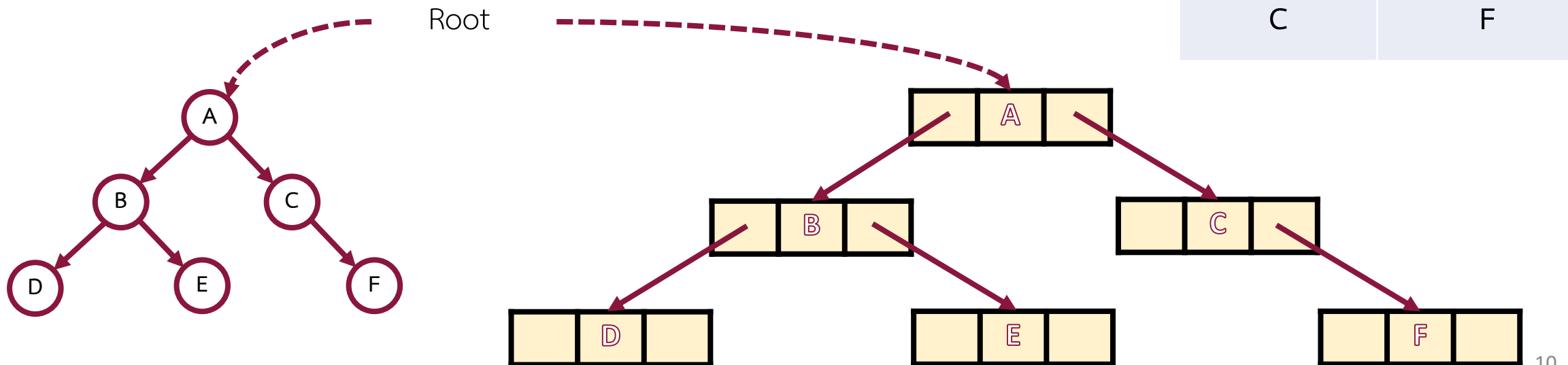
- Ancestor
  - The node from root to destination node

Ancestor	
Node	Output
F	A, C
D	A, B



- Descendant
  - The nodes are every node that inheritance from selected node.

Descendant	
Node	Output
A	B, C, D, E, F
B	D, E
C	F



# Level and Height or depth



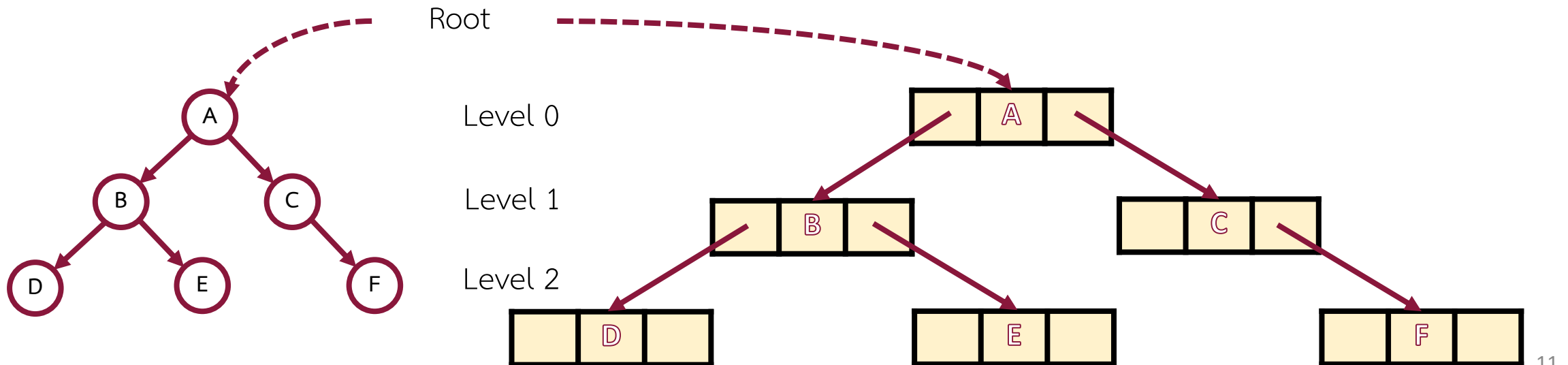
- Level

- Level is the specific level of selected node, Start at 0 as root.

- Height or depth

- Height or depth is the maximum level of binary tree. The maximum level is the highest or the deepest level + 1 due to the level starting with level 0.

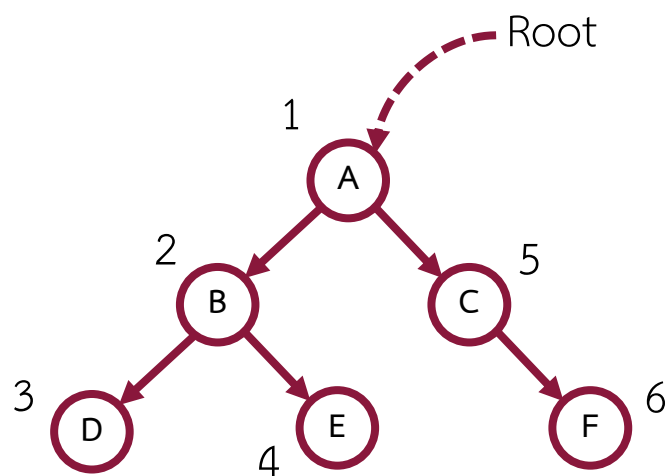
Height of depth	Maximum level + 1
3	2 + 1





- Depth first
  - Depth first traversal will check every descendant of its child before check the next child.
  - This can expand into 3 types those commonly use.
    - Preorder / Prefix
    - Inorder / Infix
    - Postorder / Postfix
- Breath first
  - Breath first will check each node by level and increase gradually until reached the highest or deepest node.

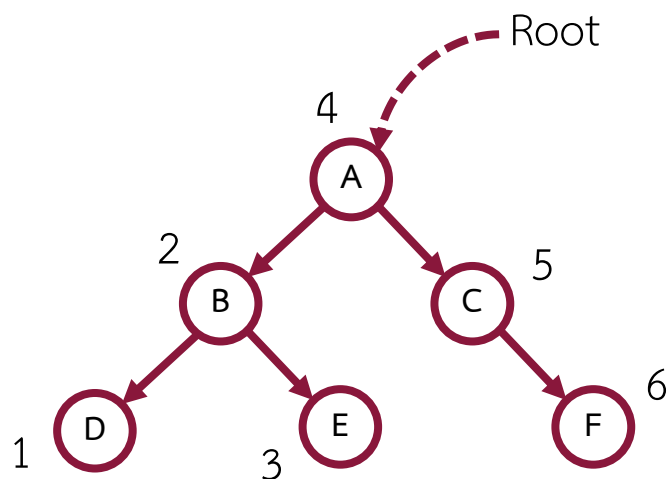
# Depth first (Pre order / Prefix)



Pre order

1	2	3	4	5	6
<u>A</u>	<u>B</u>	D	<u>E</u>	C	F

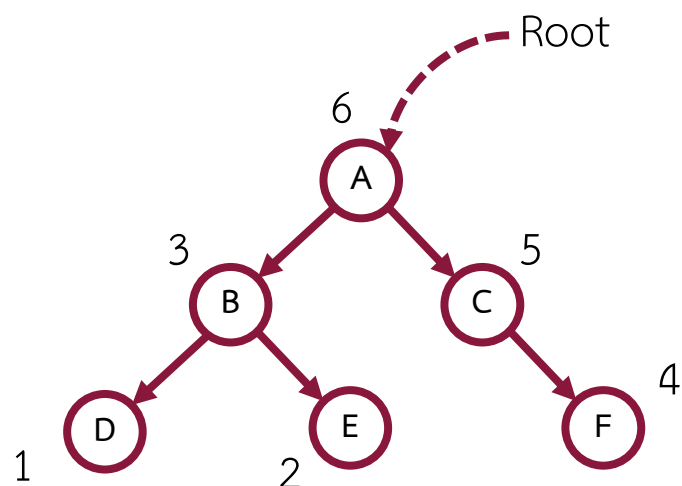
# Depth first (In order / Infix)



In order

1	2	3	4	5	6
D	<u>B</u>	E	<u>A</u>	<u>C</u>	F

# Depth first (Post order / Postfix)

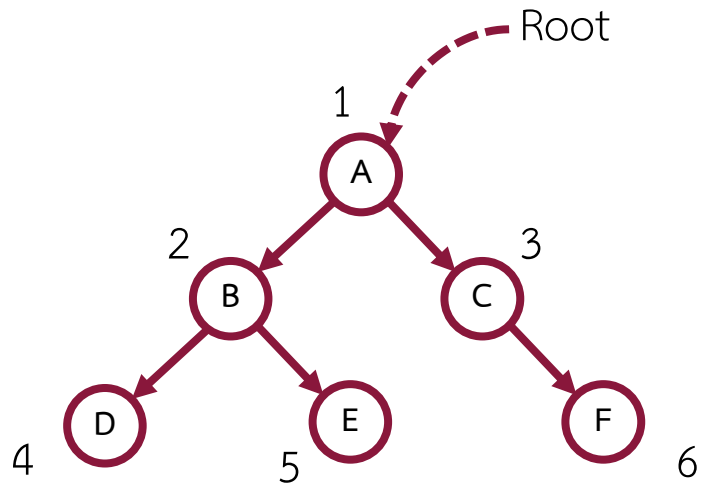


Post order

Post order					
1	2	3	4	5	6
D	E	<u>B</u>	F	<u>C</u>	<u>A</u>



# Breath first

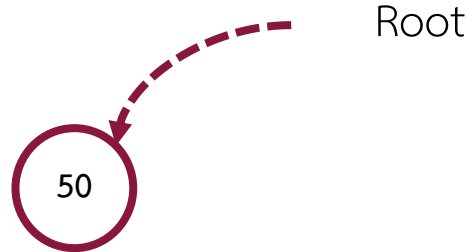


Breath					
1	2	3	4	5	6
<u>A</u>	<u>B</u>	<u>C</u>	D	E	F



# Creation and insert node into binary tree

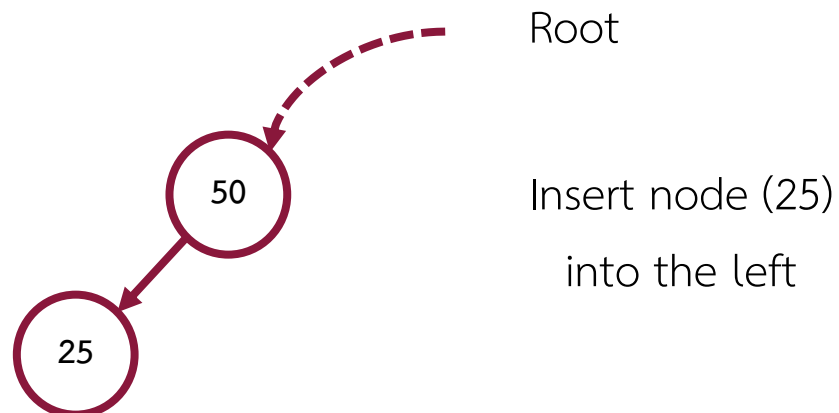
Creation node (50)  
as root



Insert  
node (25)

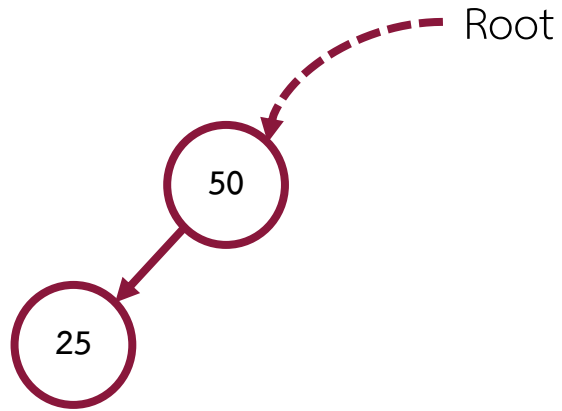


Node	Input	Comparison	Direction
50	<u>25</u>	<u>25</u> < 50	Left





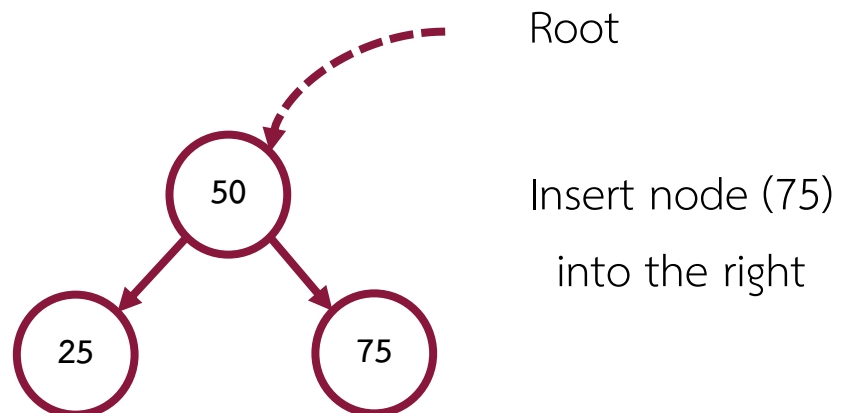
# Creation and insert node into binary tree



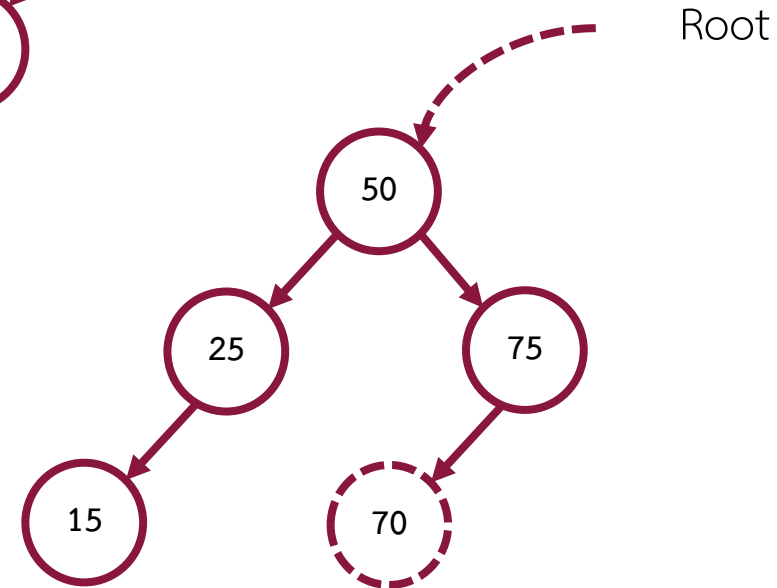
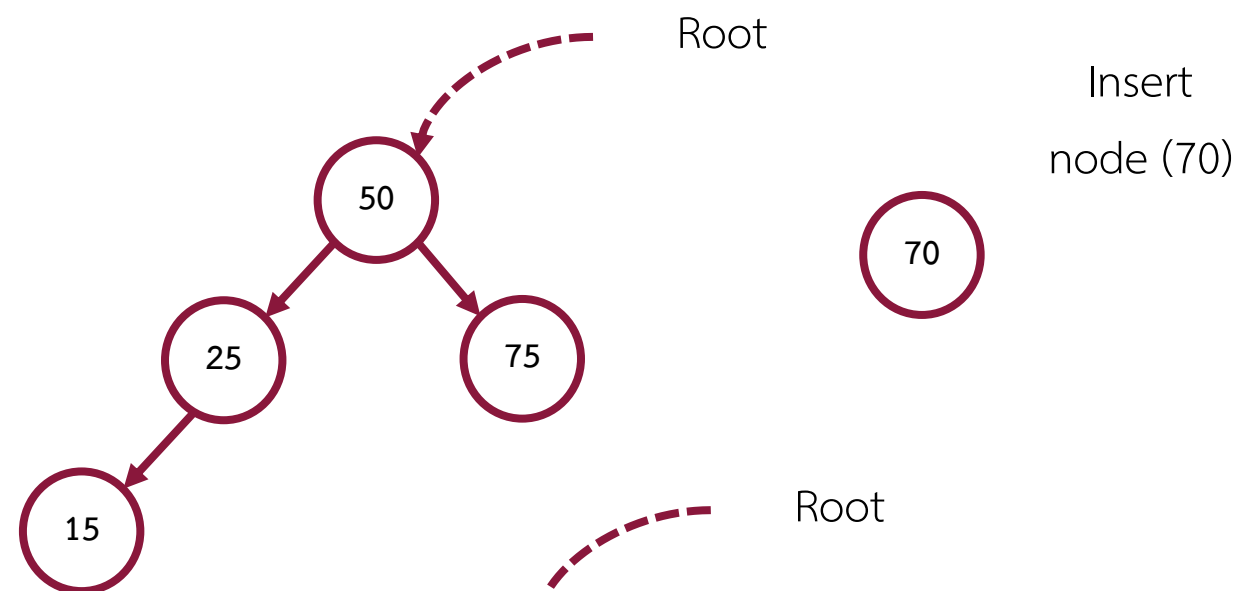
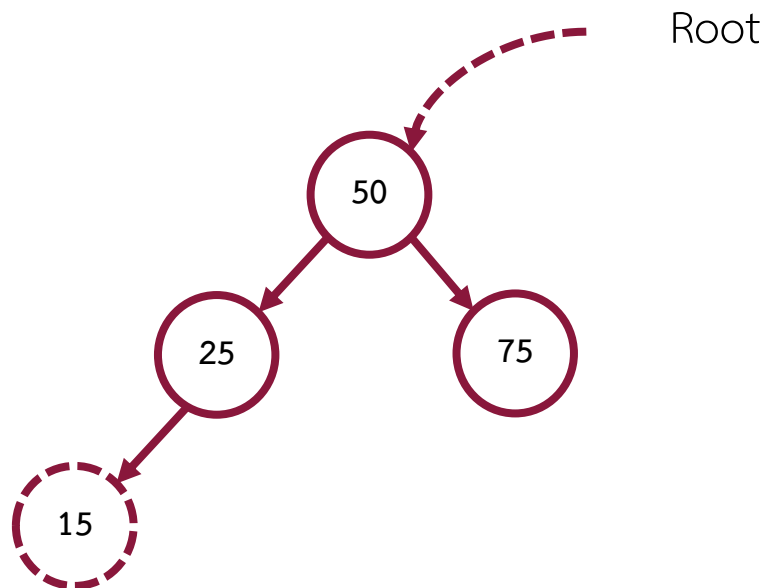
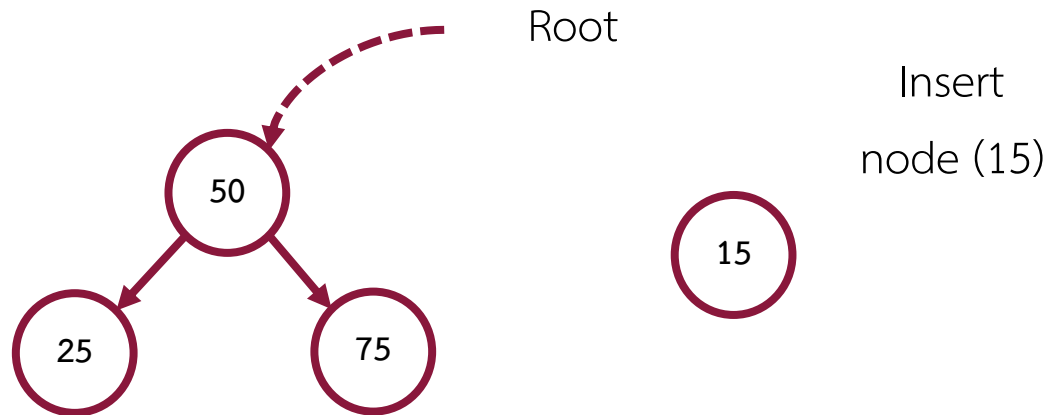
Insert  
node (75)



Node	Input	Comparison	Direction
50	<u>75</u>	$50 < \underline{75}$	Right



# Creation and insert node into binary tree

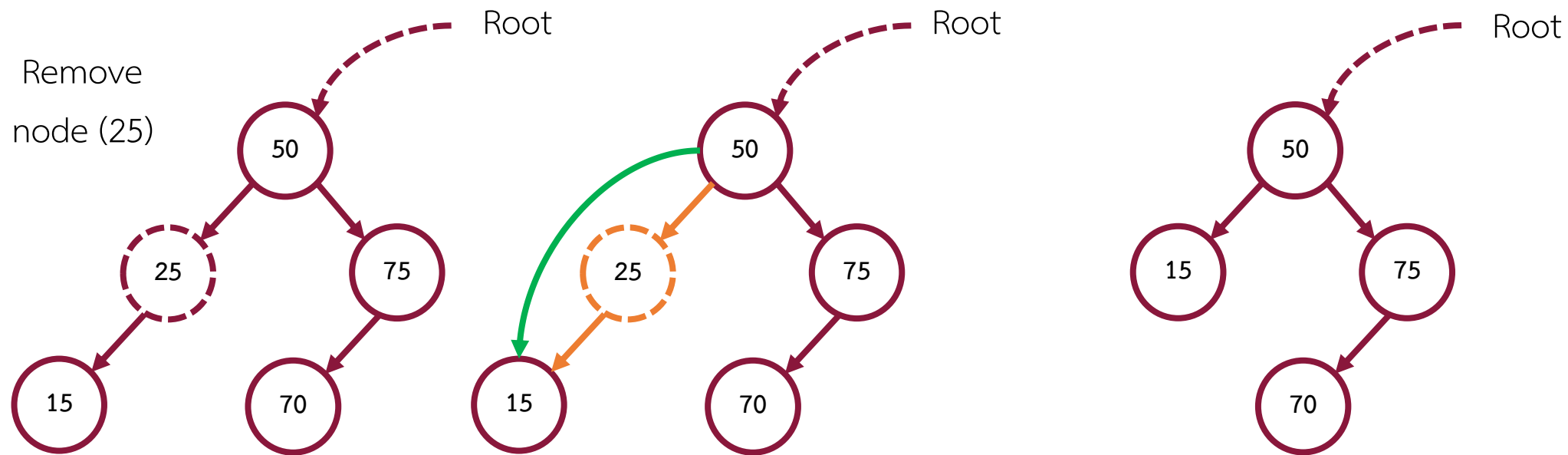




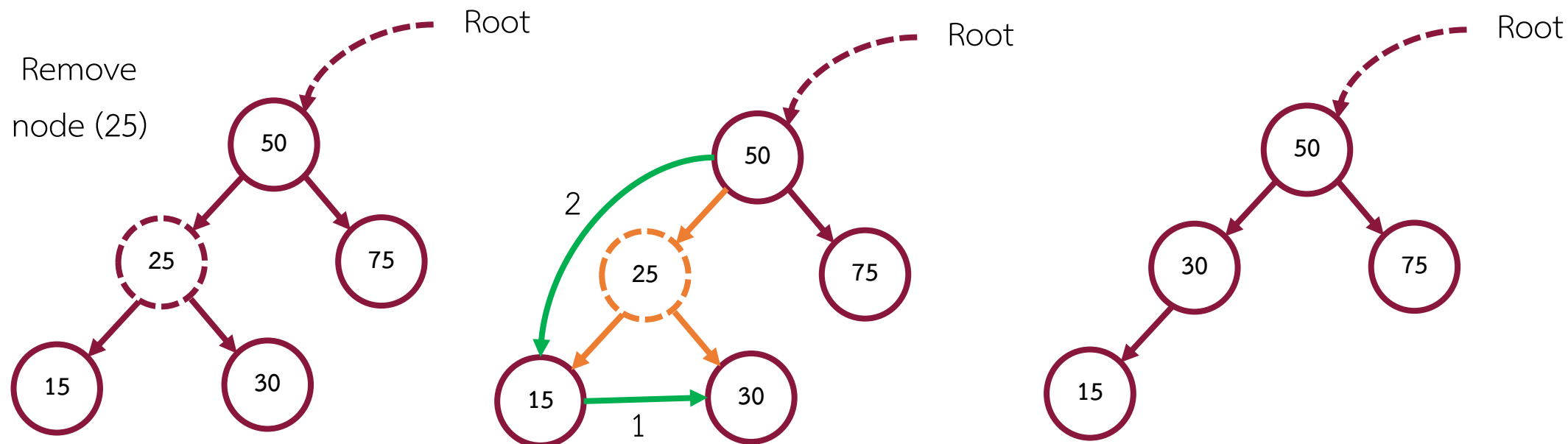
ให้นักศึกษาจับกลุ่ม 1-3 คน เขียนโปรแกรม binary tree โดยมีเงื่อนไขต่อไปนี้

- การส่งงานจะเป็นการรัน code โดยสามารถปรับแก้ไขข้อมูล input และข้อมูล node ที่จะถูกลบทิ้งได้
- โปรแกรมต้อง insert node, delete node ได้อย่างถูกต้อง และแสดงผลลัพธ์ (Result) ของ binary tree อยู่ในรูปแบบ Prefix, infix และ postfix ได้
- การให้คะแนนจะพิจารณาความสมบูรณ์ของ code ที่ส่งมาและให้นักศึกษาตอบคำถามรายบุคคล โดยส่วนของคะแนนของแต่ละคนจะไม่เท่ากันขึ้นอยู่กับคำตอบคำถาม (ในกรณีที่ตอบคำถามไม่ได้เลยจะไม่มีคะแนน ดังนั้นให้ นศ. ศึกษา coding มาให้พร้อมก่อนนัดส่ง)
- ให้นักศึกษาส่งงานล่วงหน้าวันสอบก่อน 1 วัน

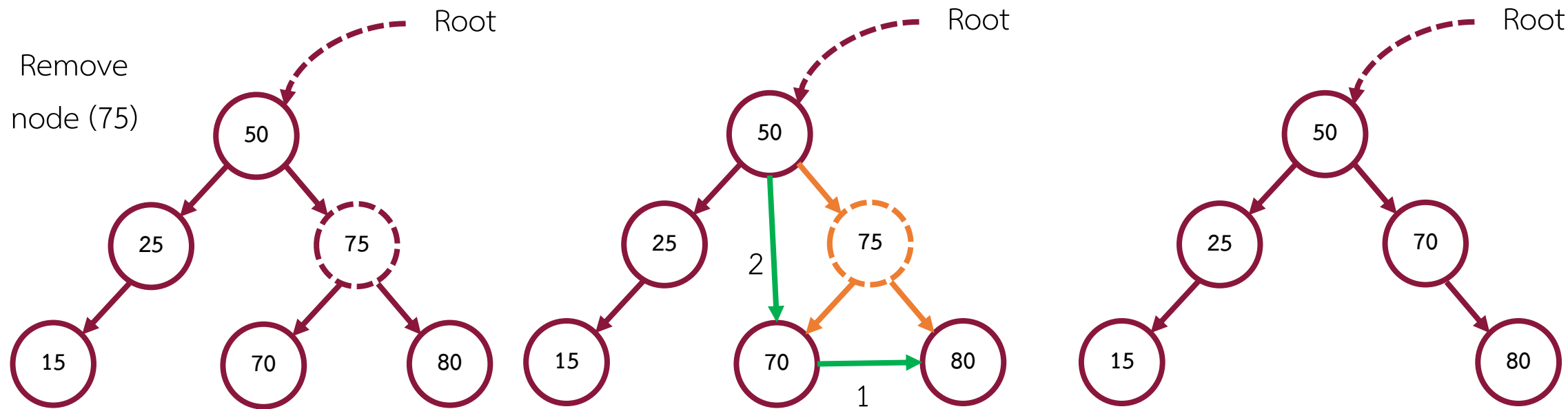
# Delete node into binary tree case (1)



# Delete node into binary tree case (2)



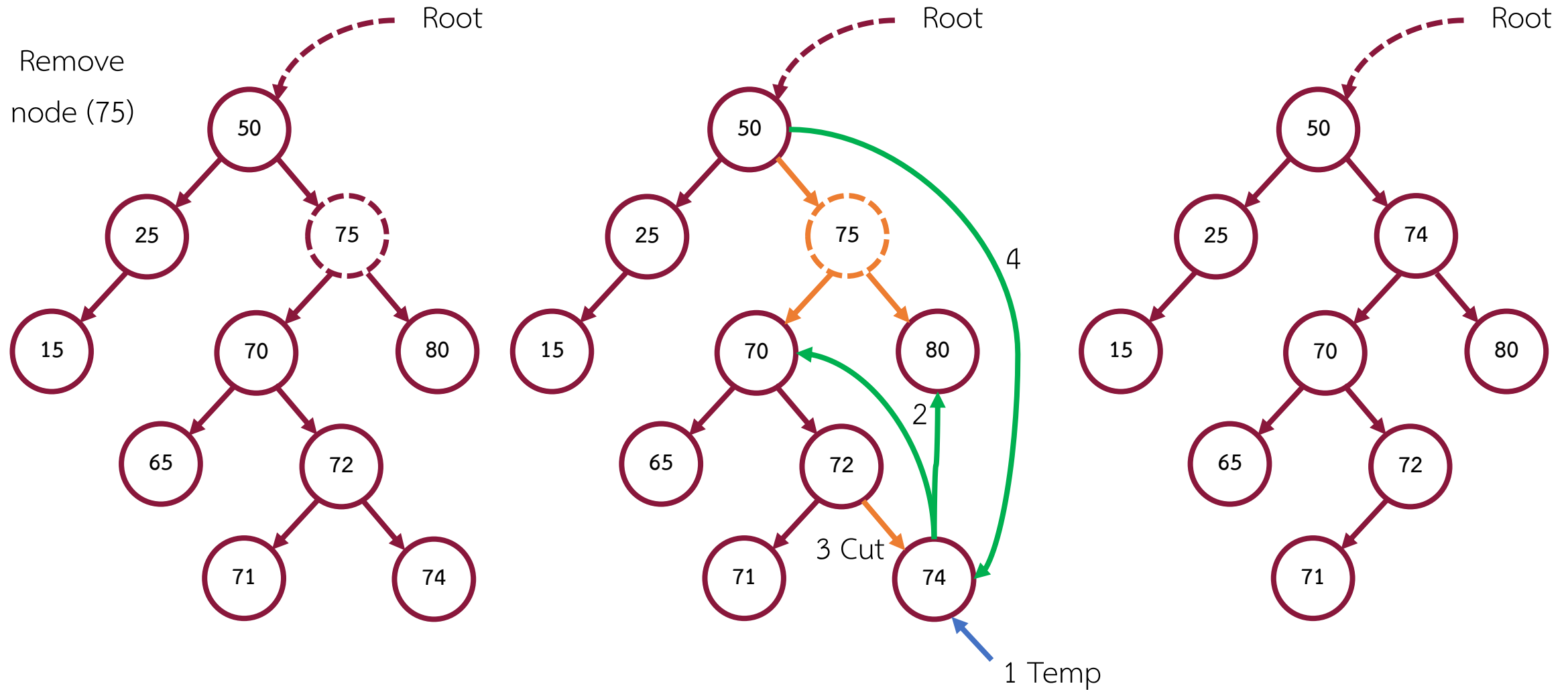
# Delete node into binary tree case (3)







# Delete node into binary tree case (4)





# Delete node into binary tree case (5)

