

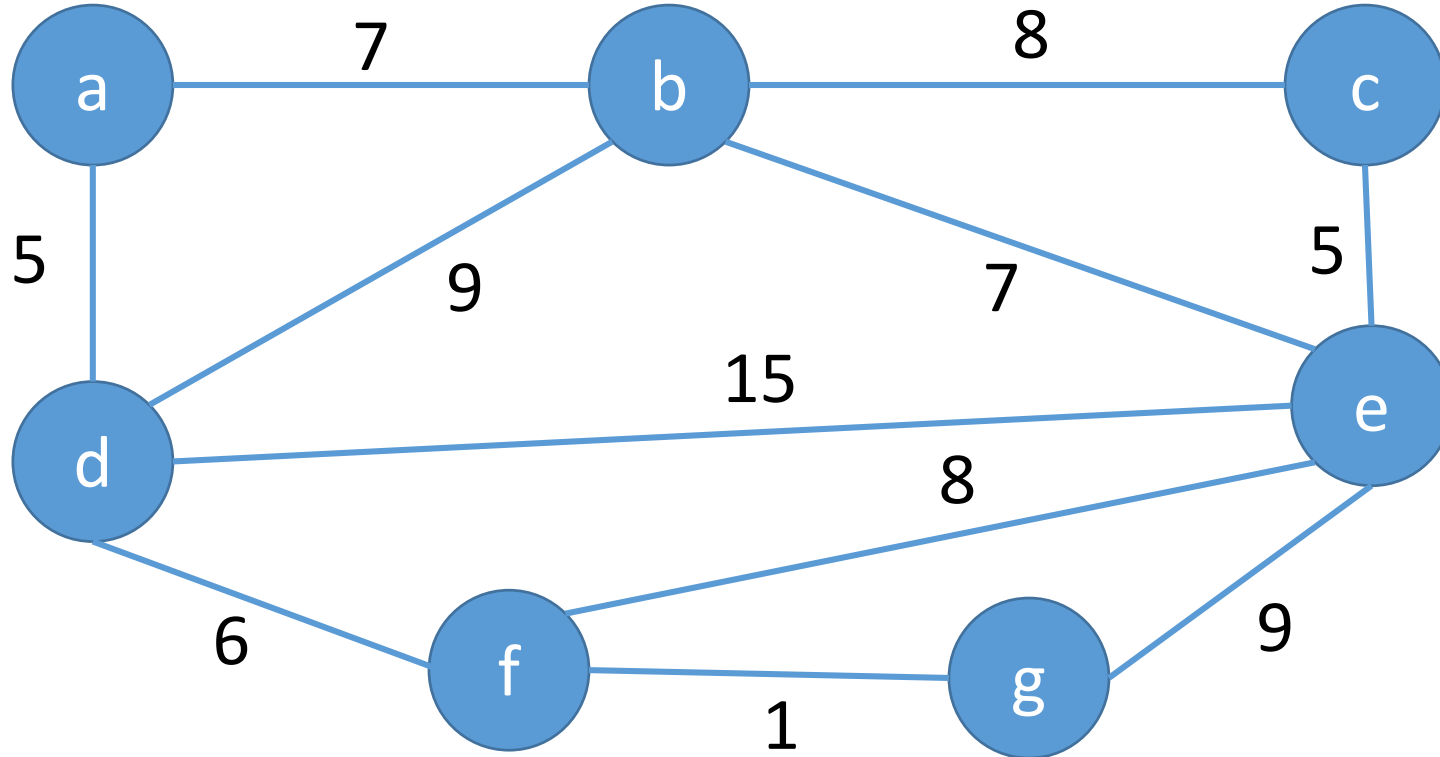
CSIT113

Problem Solving

Workshop - Week 10

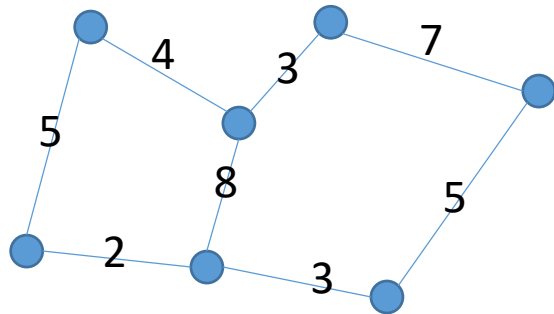
Minimal spanning Trees

- Find the minimal spanning tree of the following graph

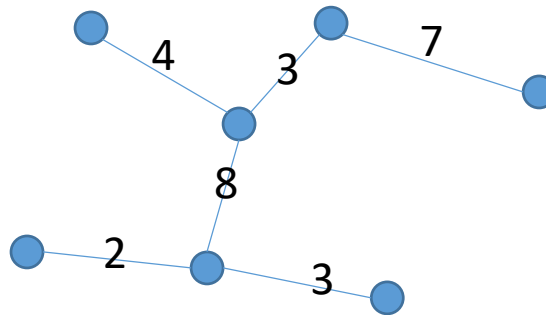


Minimal spanning tree

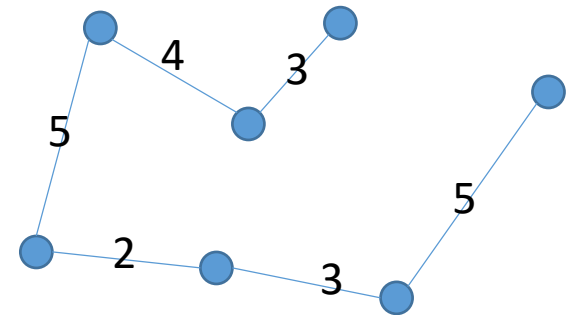
- If G is a weighted graph we can define the *weight* of a spanning tree as the sum of the weights of all the edges in the spanning tree.
- E.g.



Weight = 27



Weight = 22



- We call the spanning tree with the smallest weight the ***minimal spanning tree***.

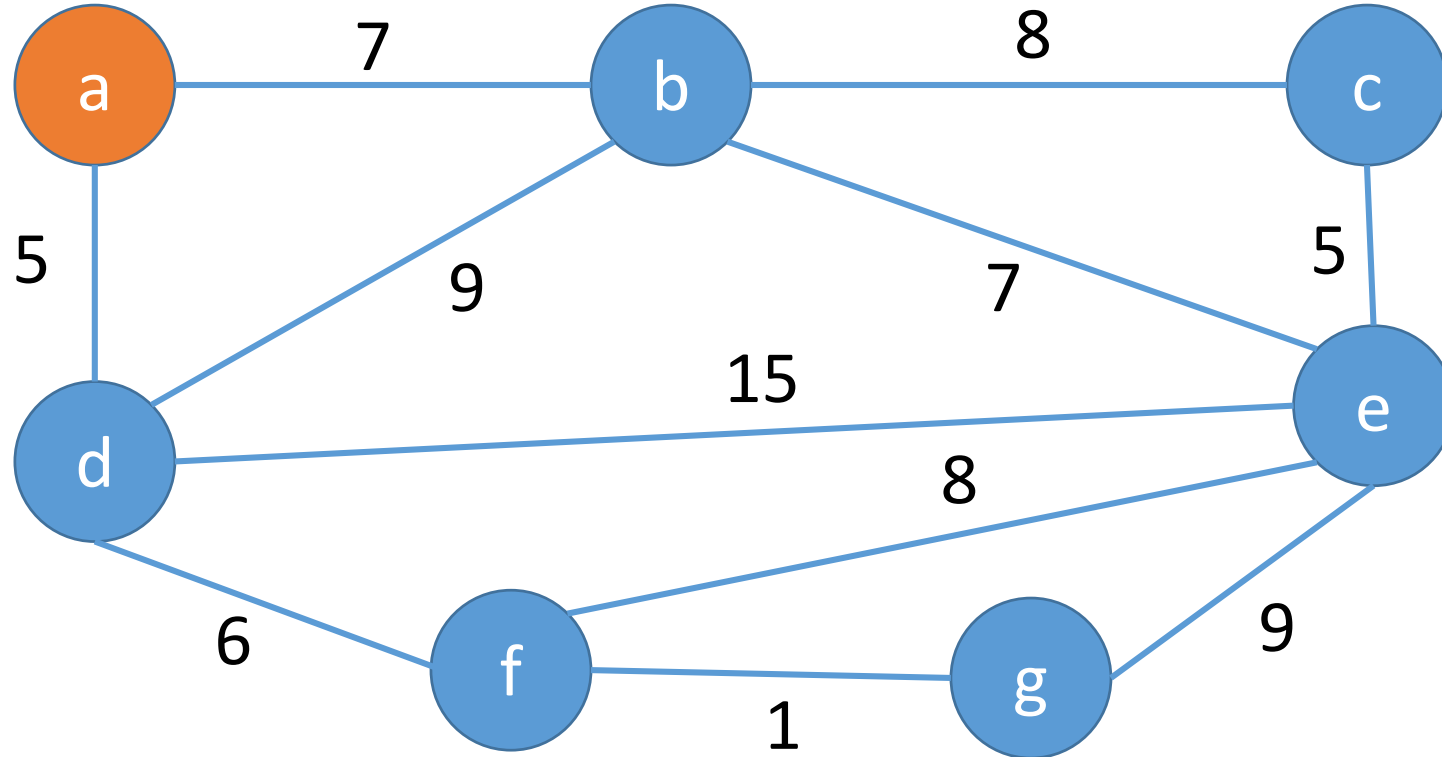
Finding the minimal spanning tree.

- If we need to find the minimal spanning tree of a graph we can take two approaches.
 - Add a nearest vertex to the spanning tree at each step.
 - Pick the smallest edge, without forming a cycle, at each step.

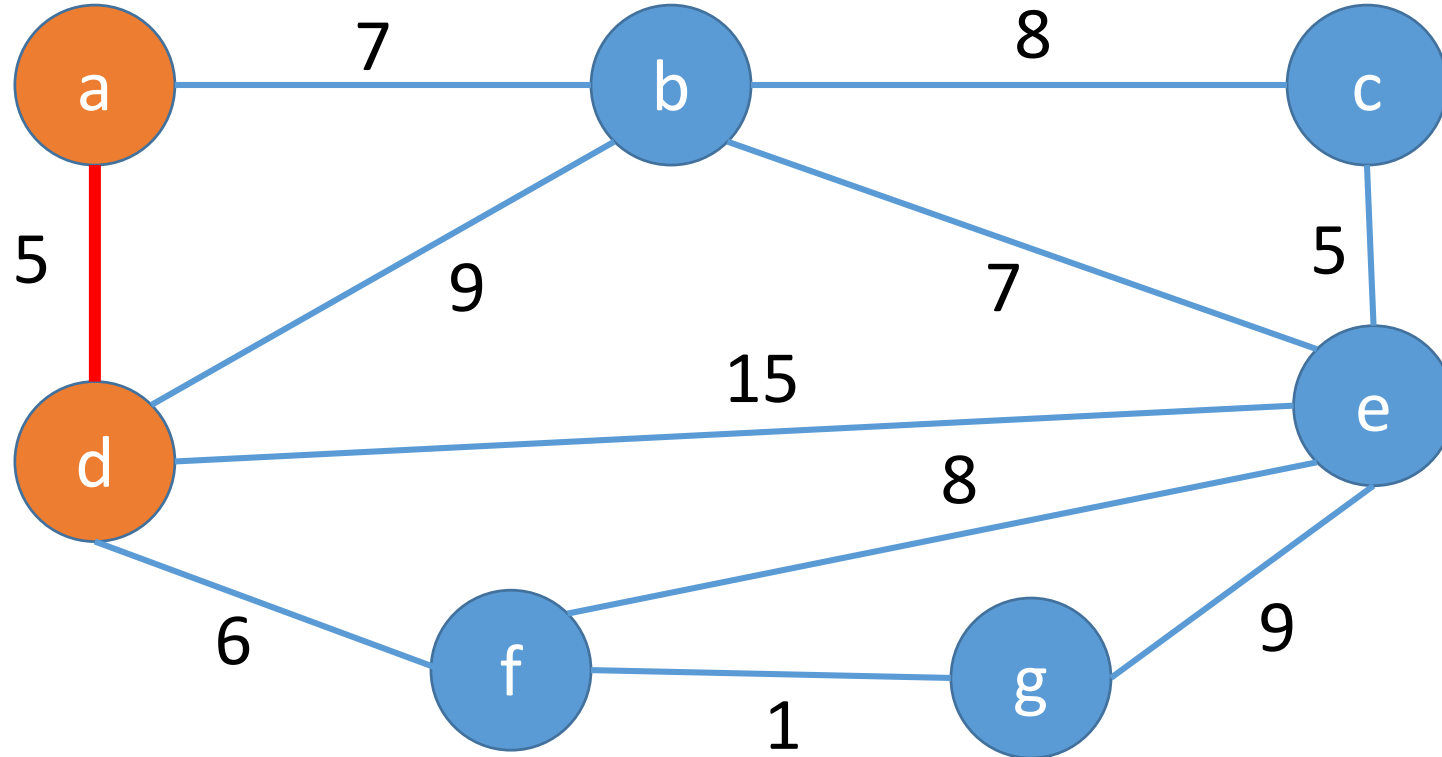
Vertex at a time (Prim's Algorithm)

- Start with any vertex. This is our starting tree.
- Find the vertex connected to the tree so far by the shortest edge.
- Add this edge and vertex to the tree.
- Repeat this process, always choosing a vertex which is not already part of the tree.
- Stop when all vertices are selected.

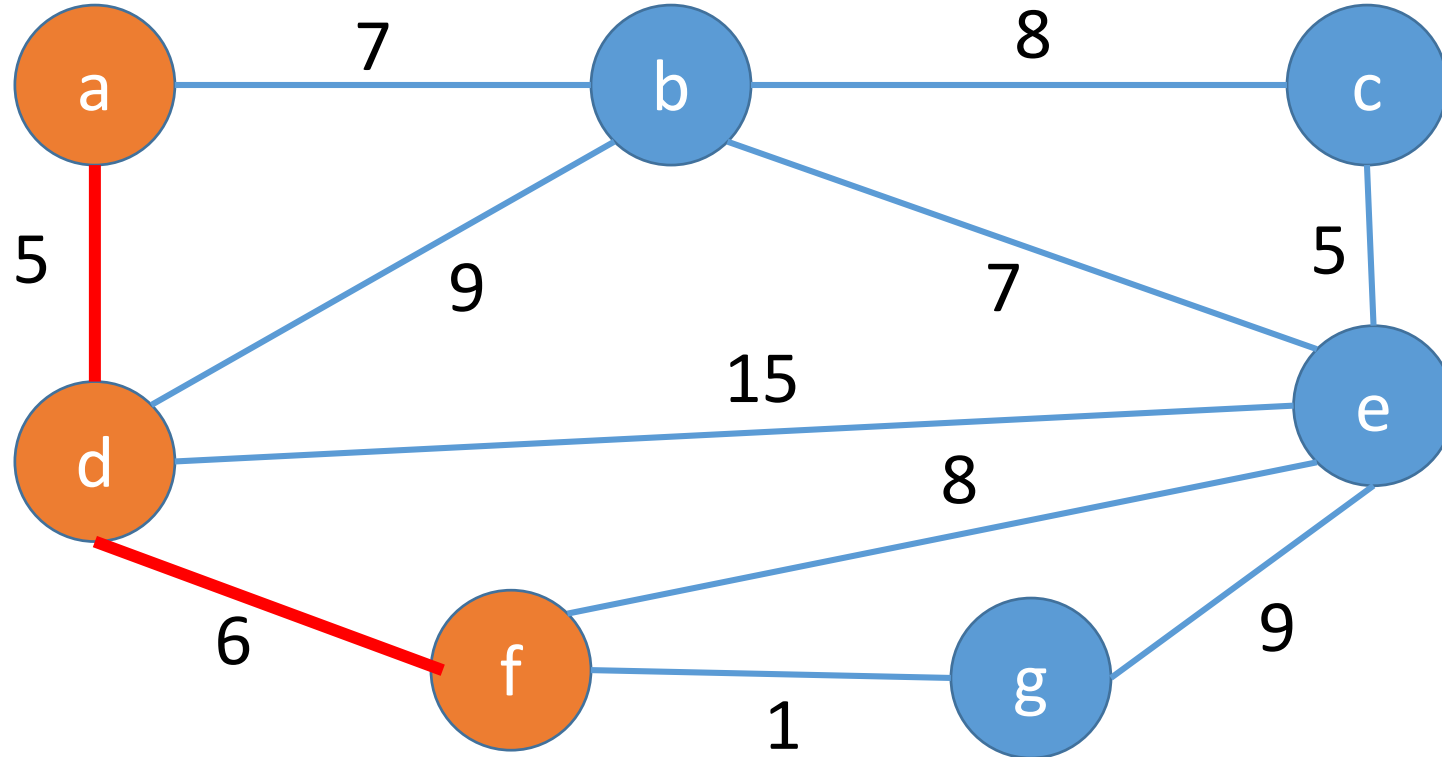
Prim's Algorithm



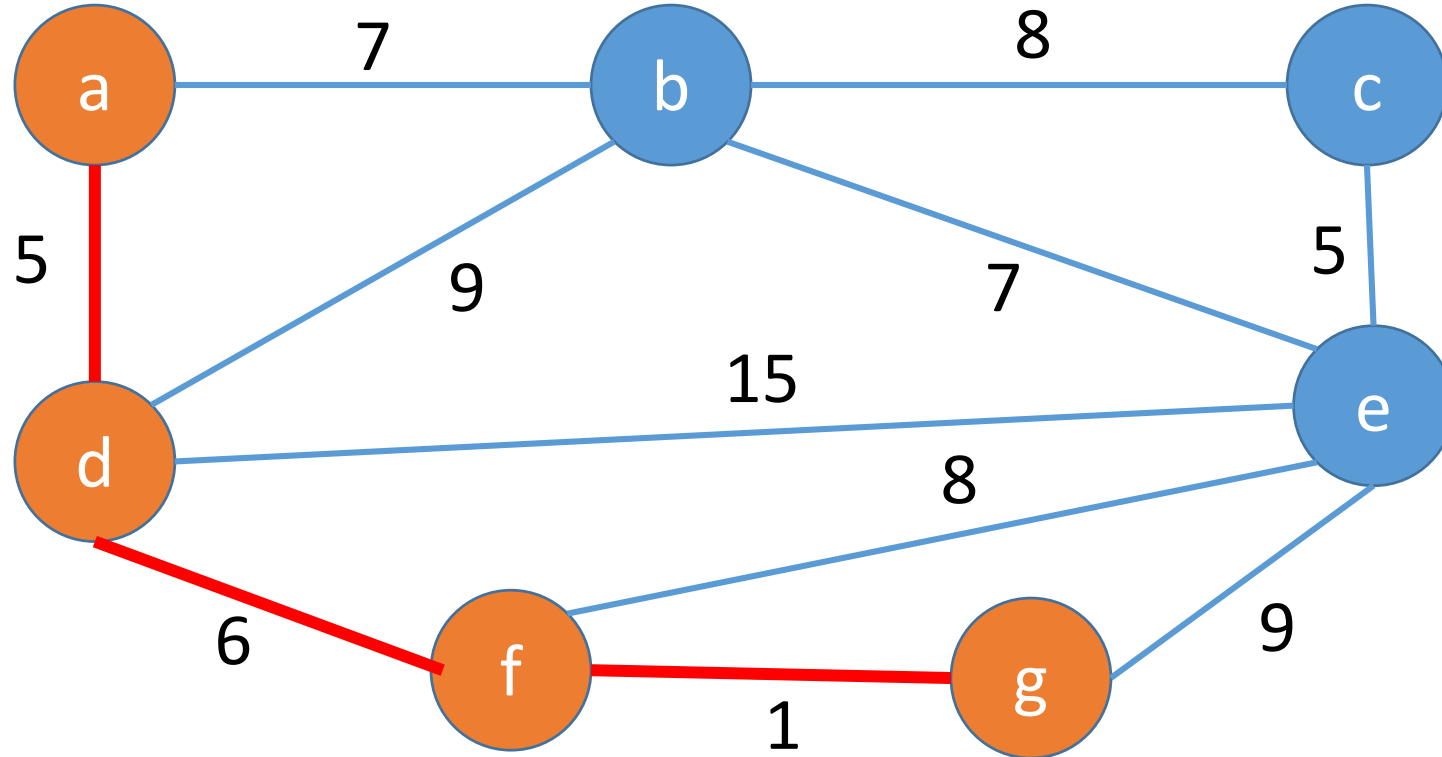
Prim's Algorithm



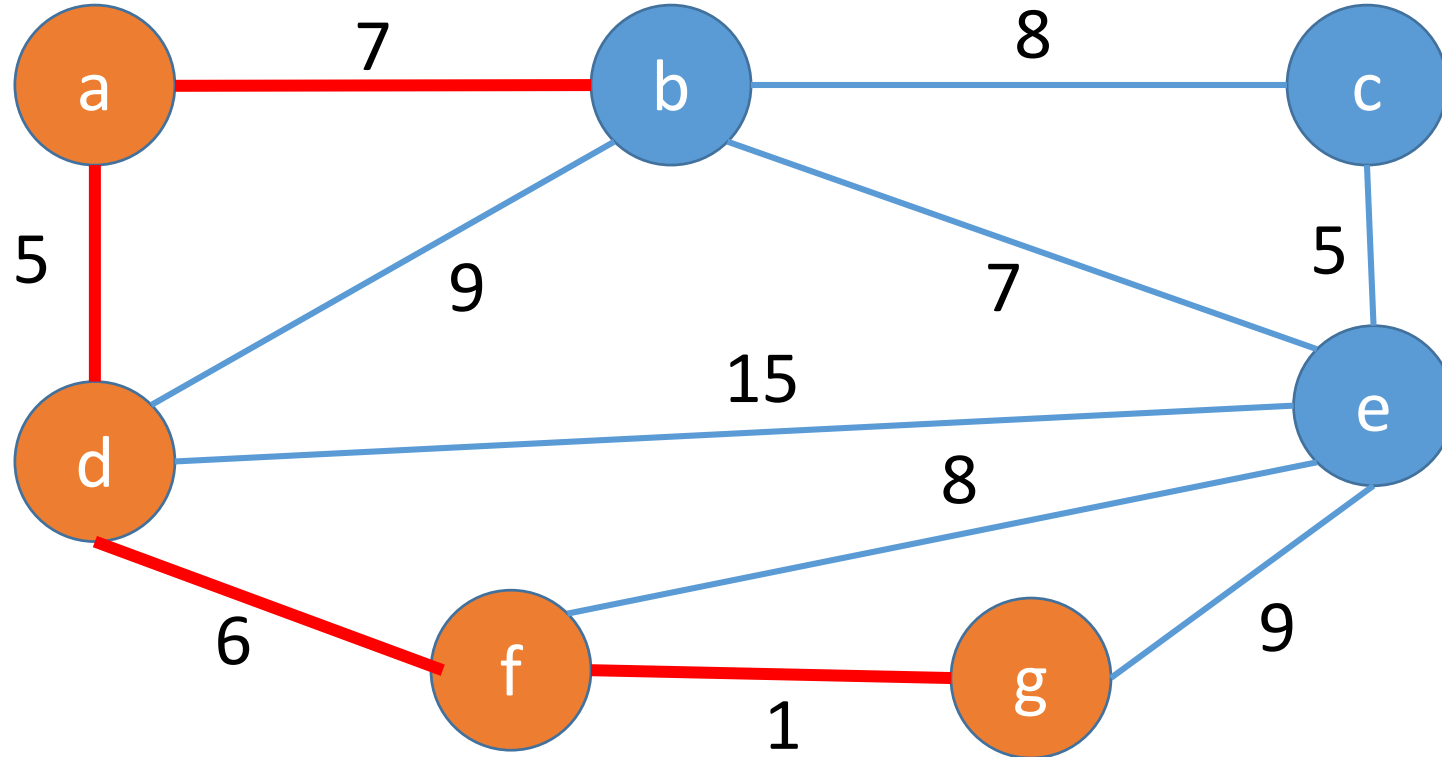
Prim's Algorithm



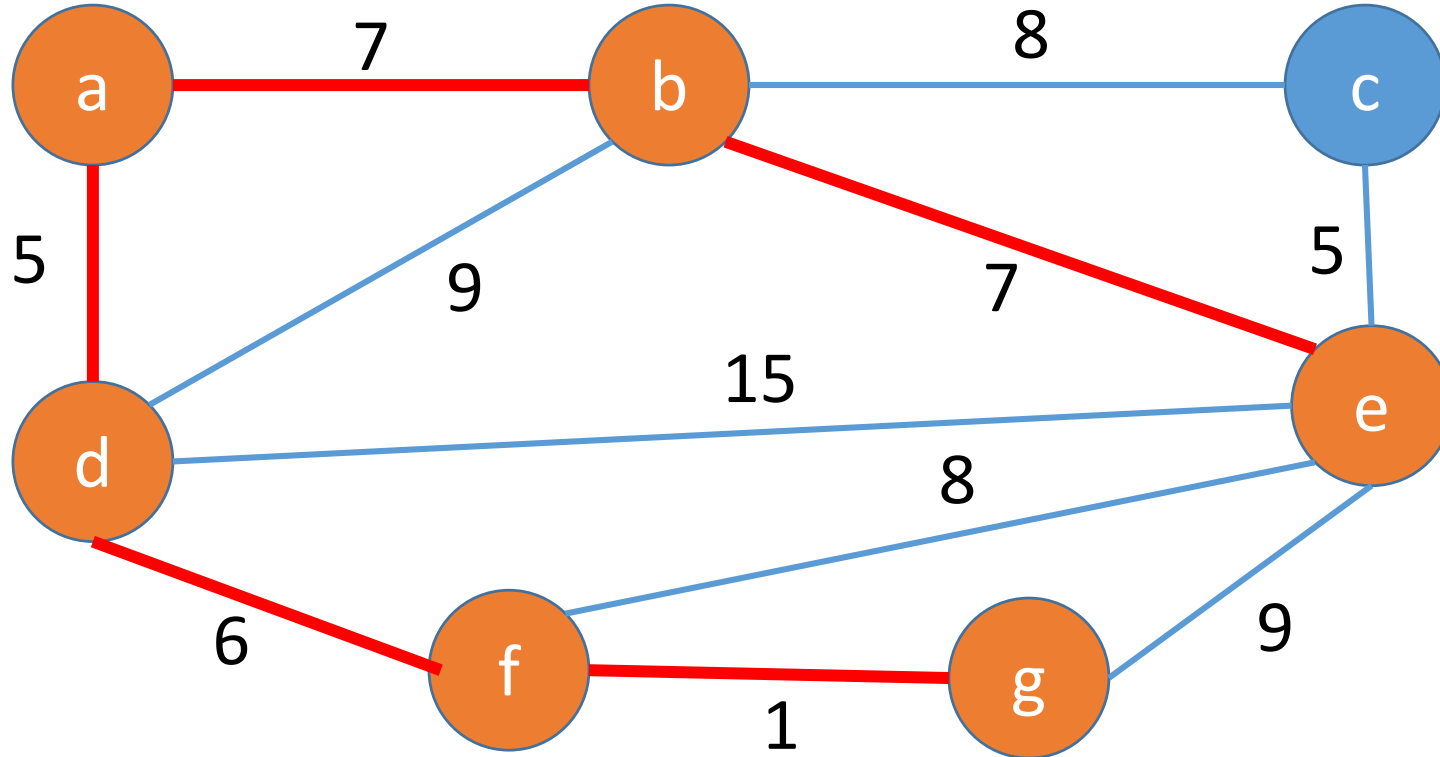
Prim's Algorithm



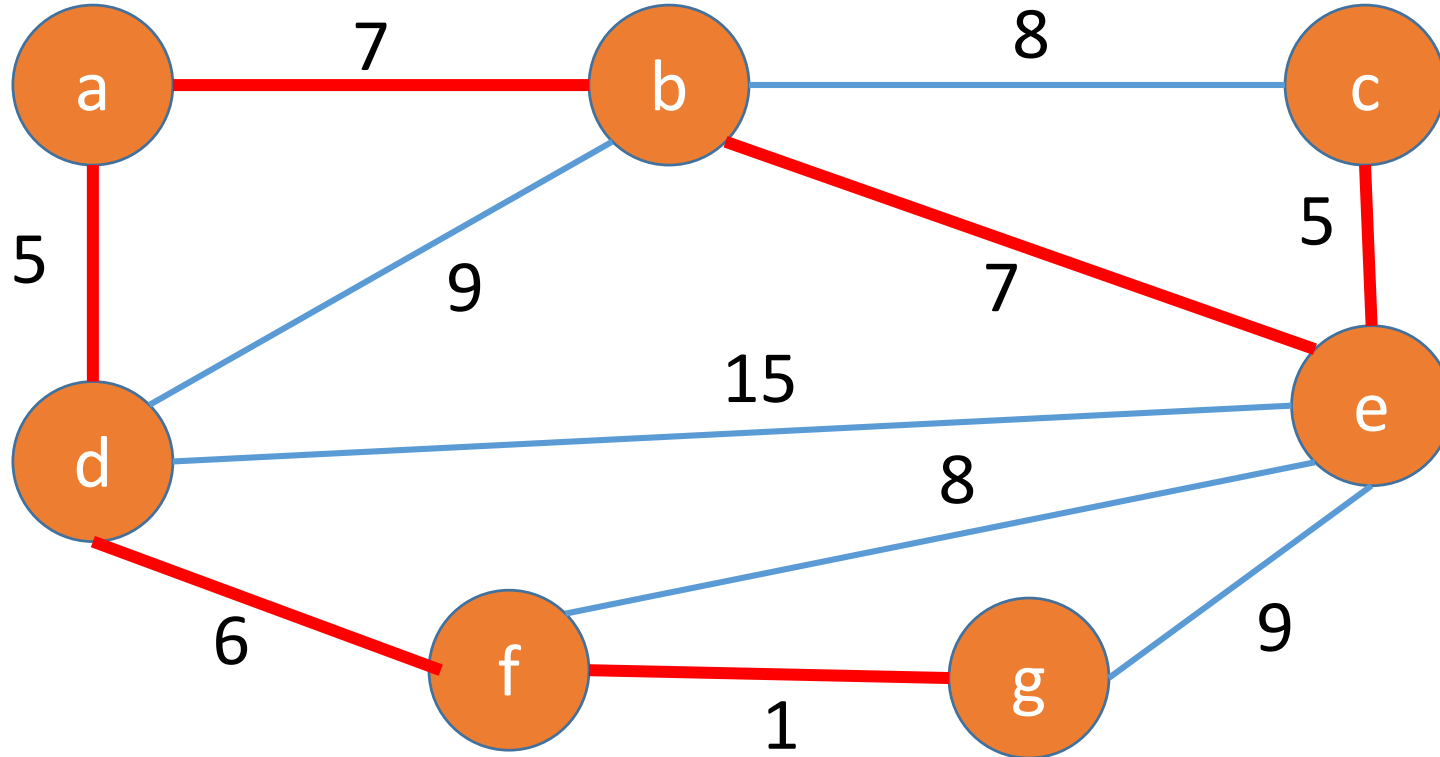
Prim's Algorithm



Prim's Algorithm



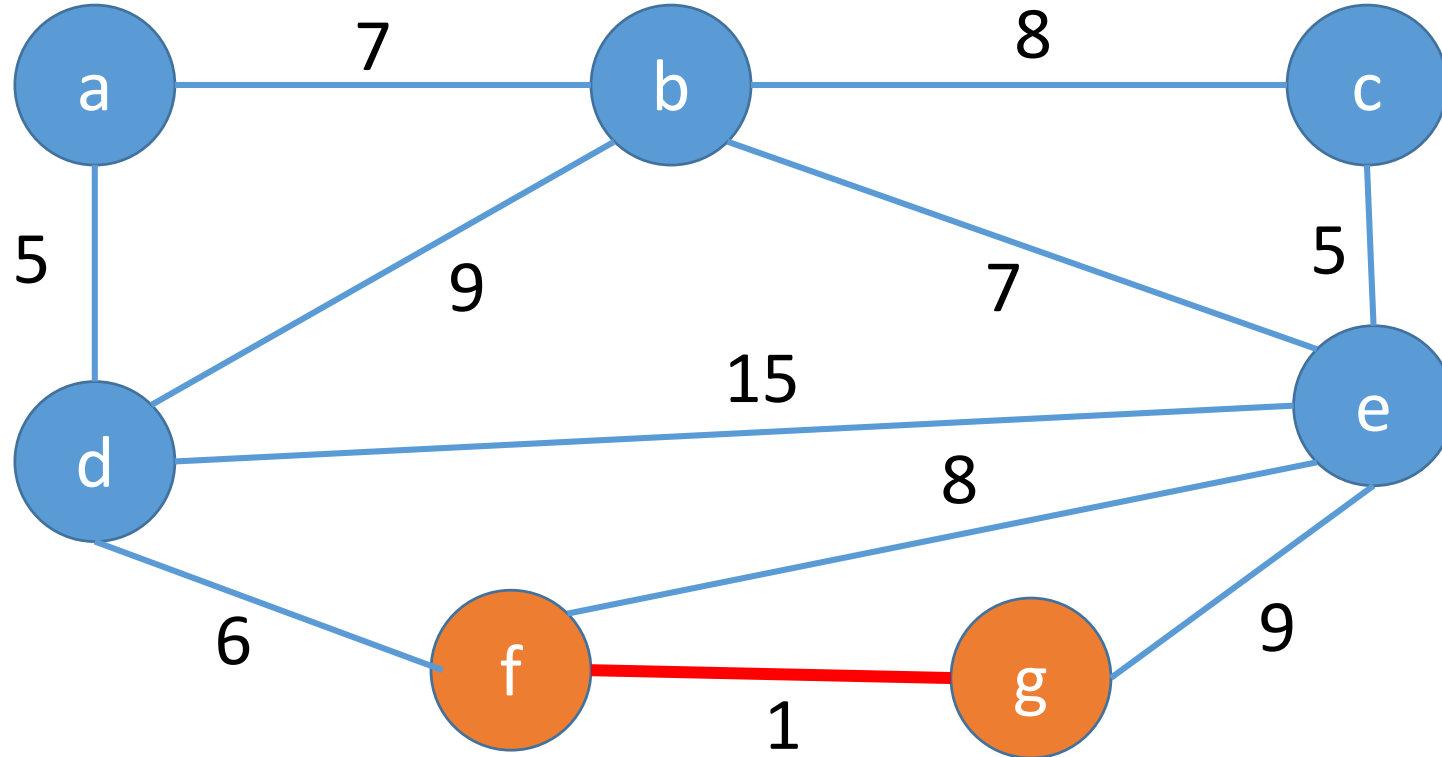
Prim's Algorithm



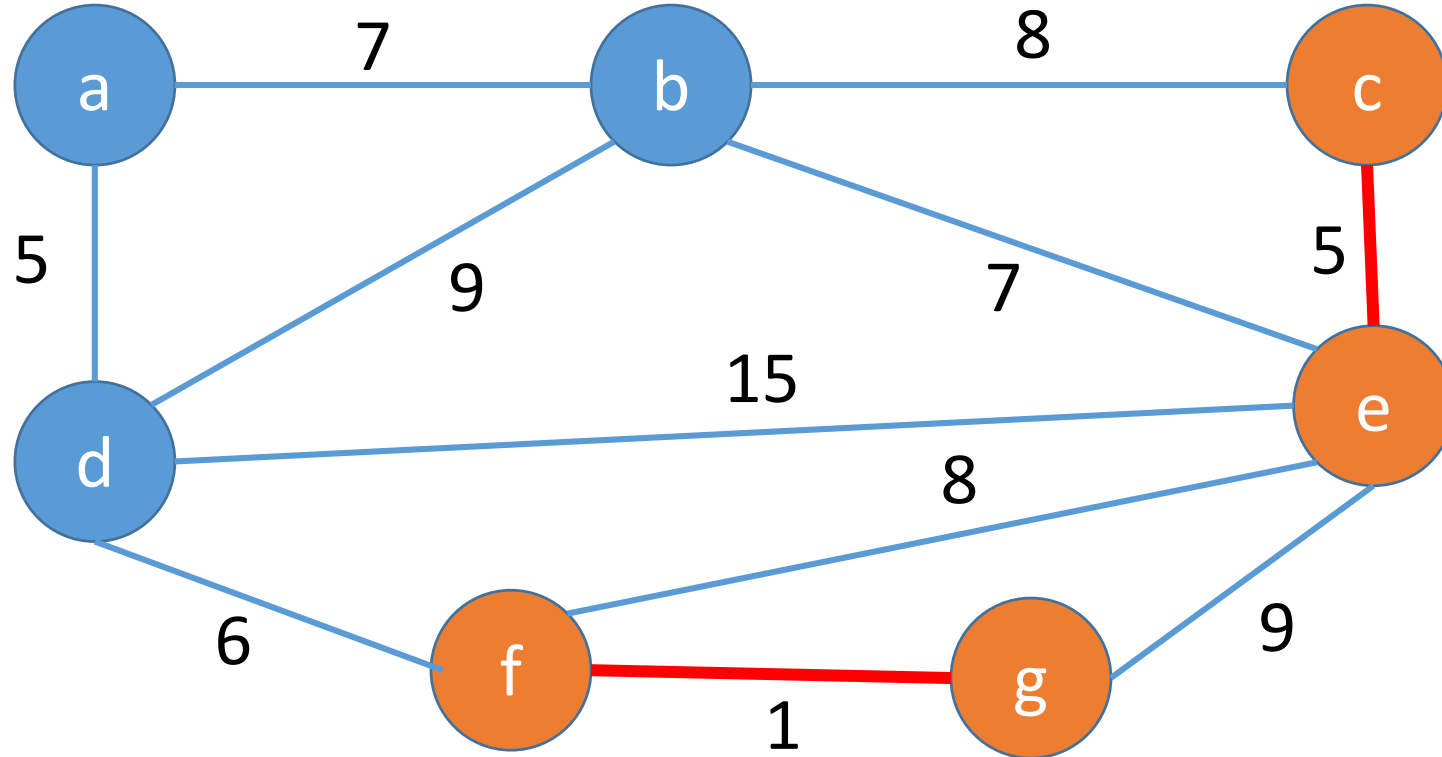
Edge at a time. (Kruskal's Algorithm)

- Pick the shortest edge. This is the first edge in our tree.
- Now consider the next shortest edge.
 - If it connects vertices which are not already connected in a tree, pick it.
 - Otherwise reject it.
- Repeat until all vertices are connected.

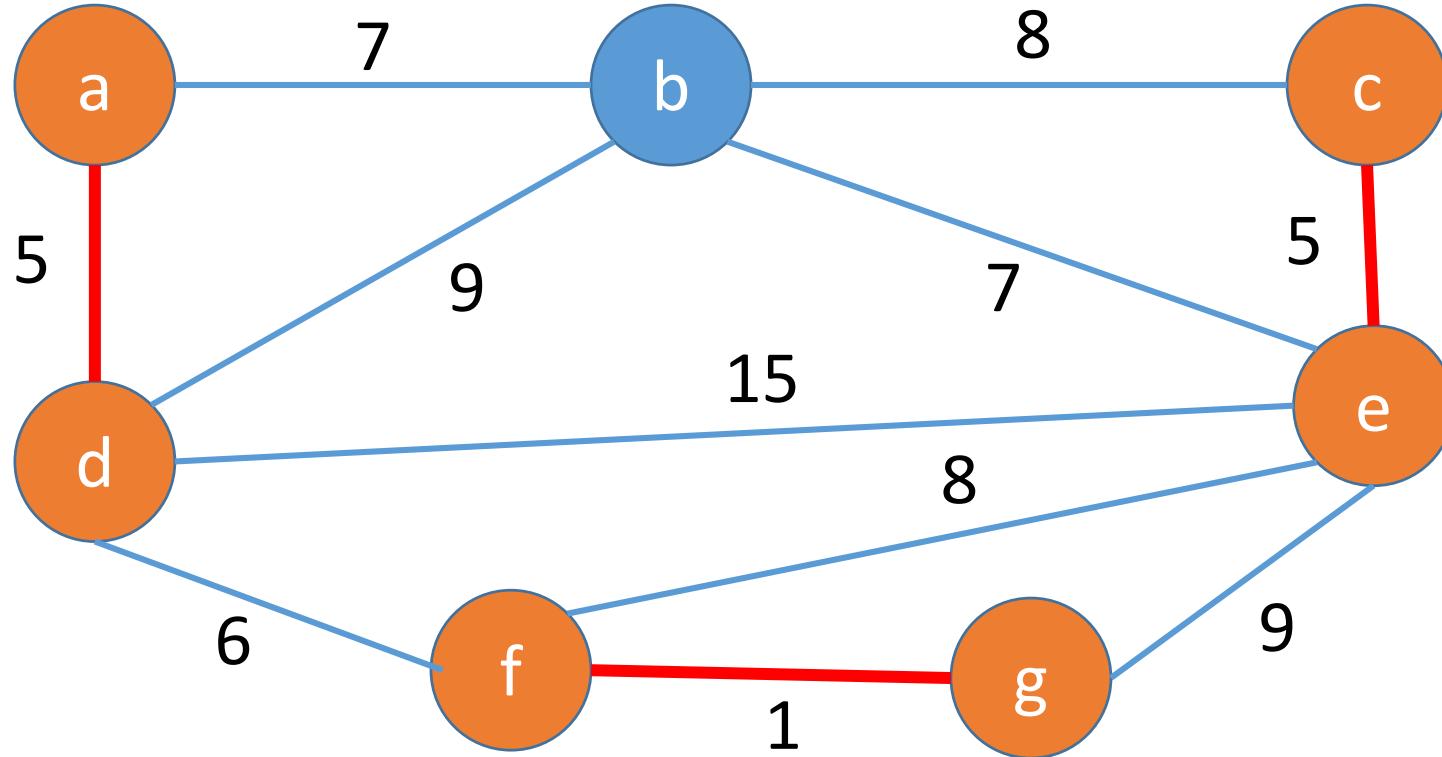
Kruskal's Algorithm



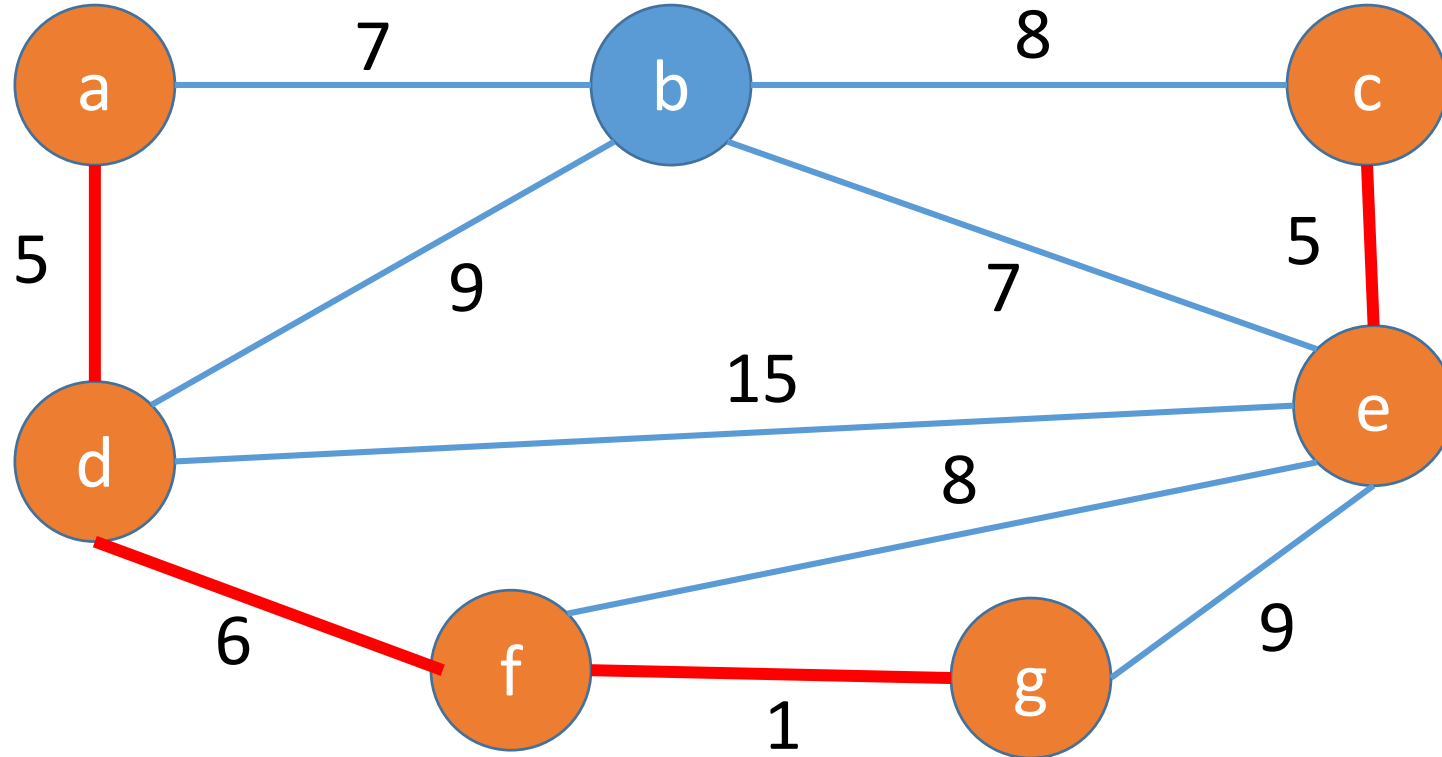
Kruskal's Algorithm



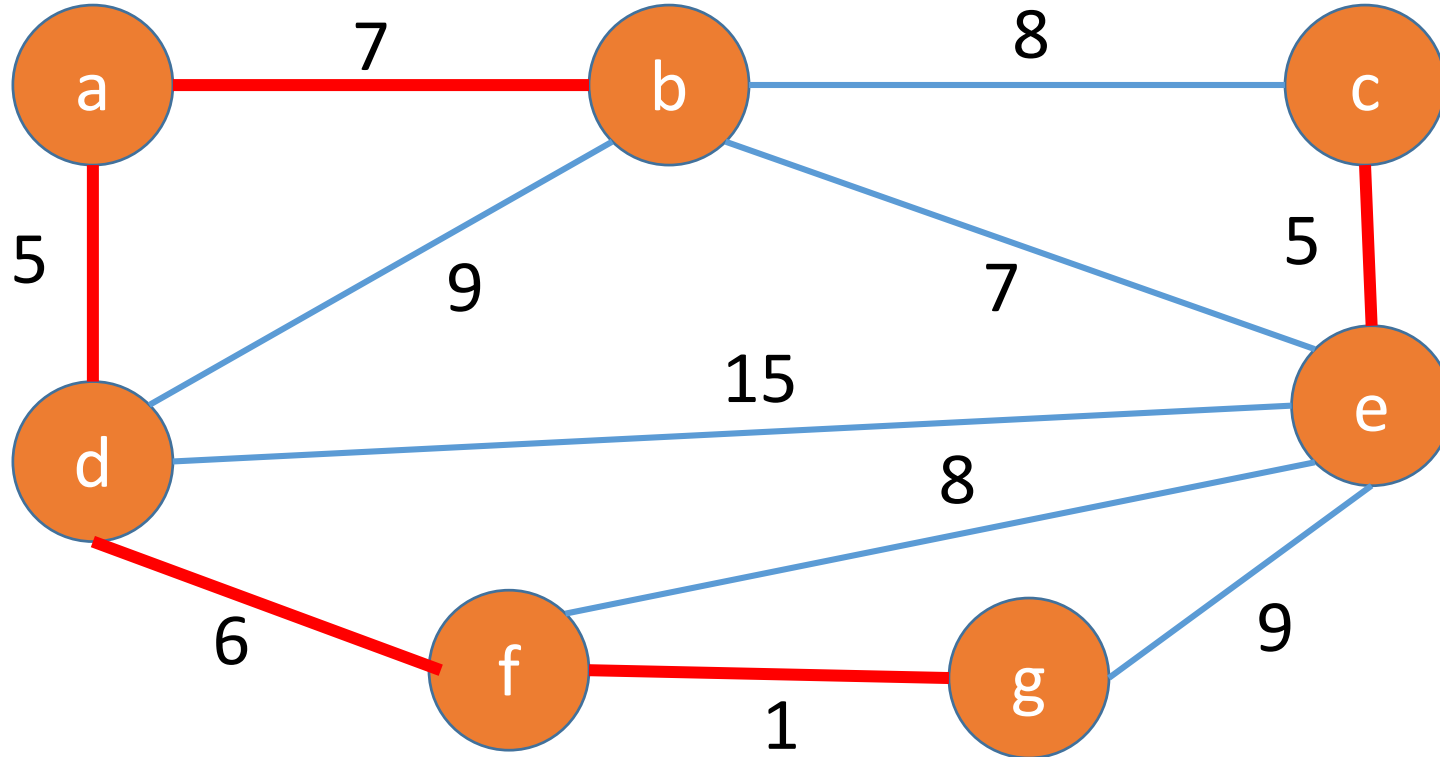
Kruskal's Algorithm



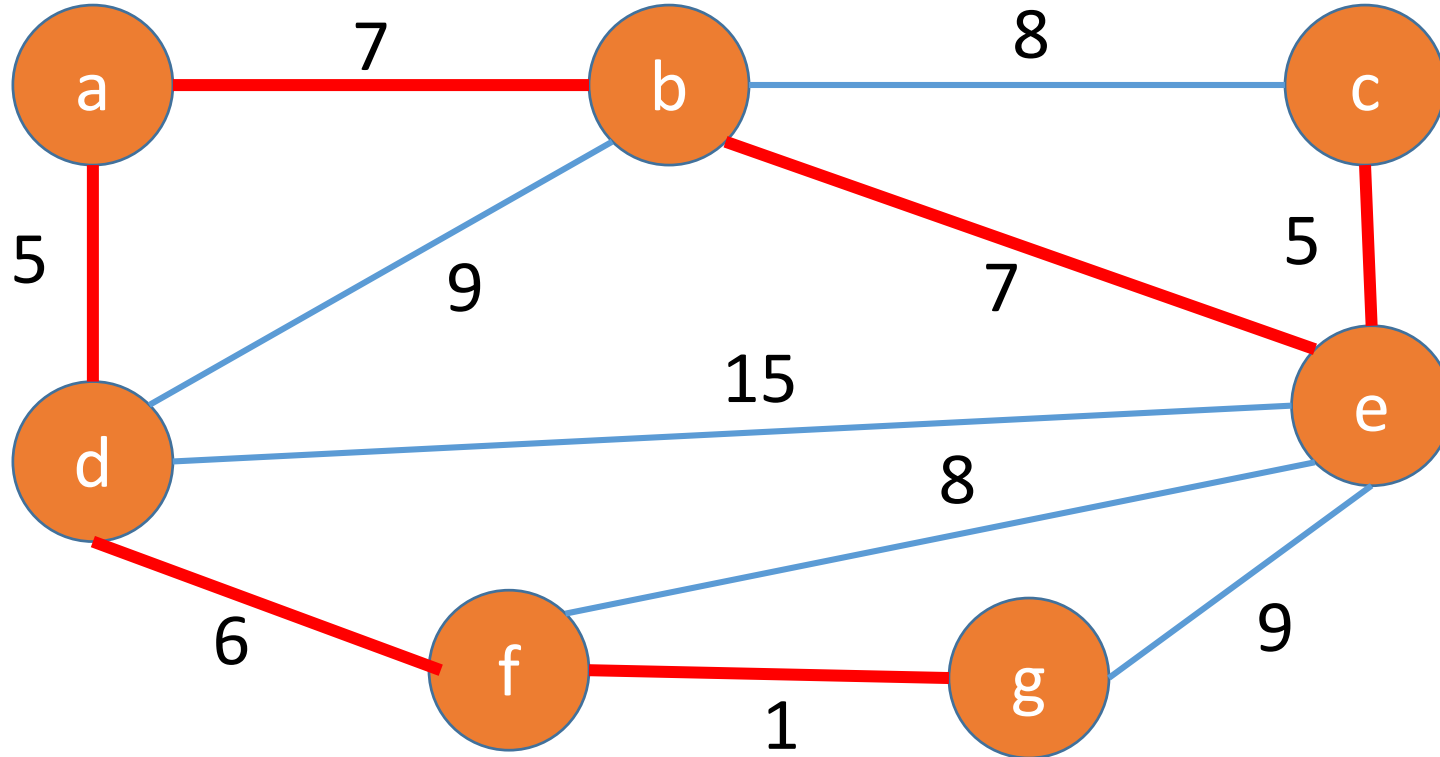
Kruskal's Algorithm



Kruskal's Algorithm

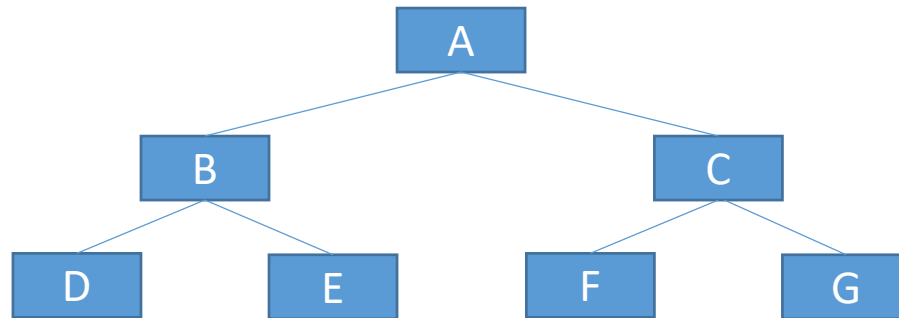


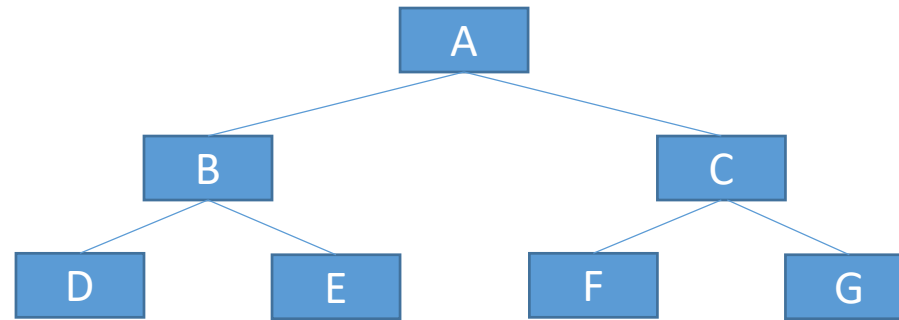
Kruskal's Algorithm



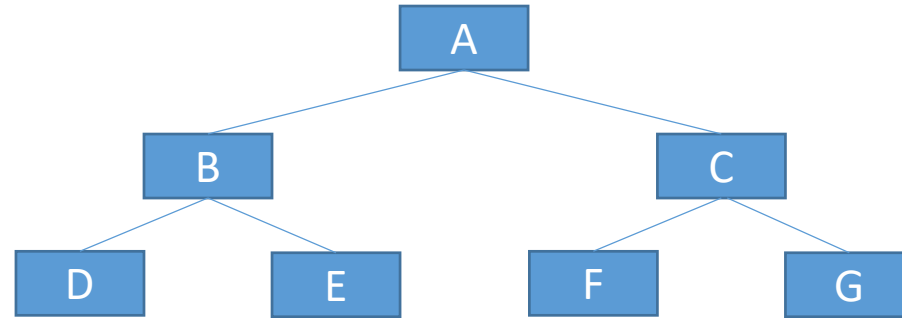
Listing Trees

- Sometimes we need to do something to every item in a tree.
- Trees are not ideally arranged for this process.
 - They are better suited for taking a single path from root to leaf.
- How can we efficiently list all the elements in a tree?
- E.g.

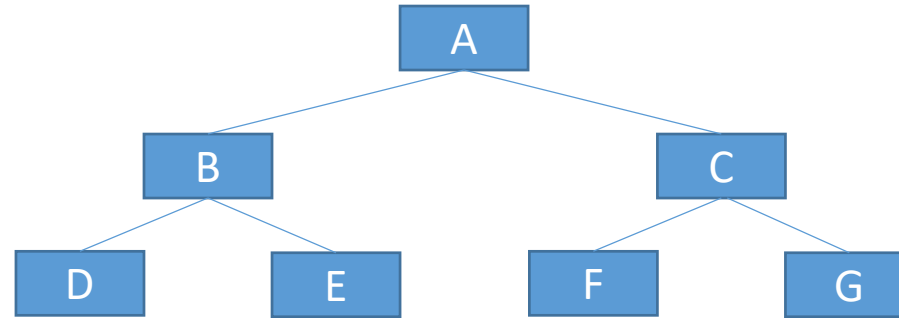




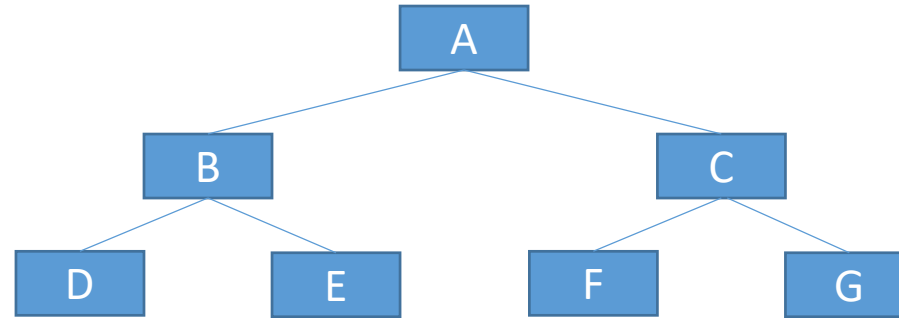
- The obvious list that this tree suggests is [A, B, C, D, E, F, G]
- How do we get this?
- Easy! List all of the nodes, left to right, at each level.
- This called a ***breadth-first*** traversal
- Actually, Not easy! The tree can only be navigated via the edges.
- This means that we are constantly travelling up and down branches and have to remember all the places we have been so far.



- The alternative is ***depth-first*** traversal.
- With this approach we go down a branch to the leaf before we traverse the rest of the tree.
- We can easily achieve this with a recursive procedure; **Visit**.
- Visit (node)
 - Visit (left child)
 - Visit (right child)
- We can see this working on the next slide.



- Visit (A)
 - Visit (B)
 - Visit (D)
 - Visit (E)
 - Visit (C)
 - Visit (F)
 - Visit(G)
- This traverses the tree but we still have one problem.
- When do we list the contents of the node?



- We can modify Visit to list the tree by adding a Print (node) to it.
- We can do this in any one of three locations.

Visit_pre (node)

Print (node)

Visit_pre (left child)

Visit_pre (right child)

Visit_in (node)

Visit_in (left child)

Print (node)

Visit_in (right child)

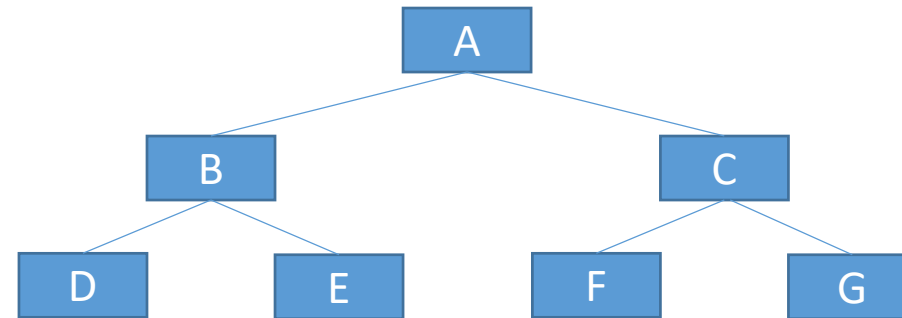
Visit_post (node)

Visit_post (left child)

Visit_post (right child)

Print (node)

- These are called ***pre-order***, ***in-order*** and ***post-order*** traversals respectively.
- Each gives us a list of the nodes in a different order.

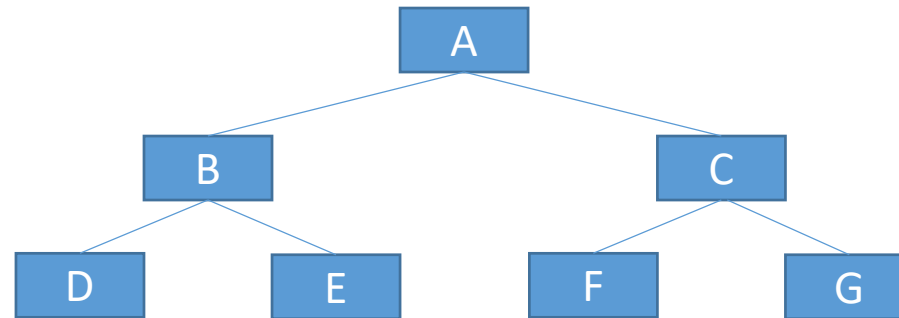


- What output do we get if we list the above tree:
 - Using pre-order traversal?
 - Using in-order traversal?
 - Using post-order traversal?

Visit_pre (node)
Print (node)
Visit_pre (left child)
Visit_pre (right child)

Visit_in (node)
Visit_in (left child)
Print (node)
Visit_in (right child)

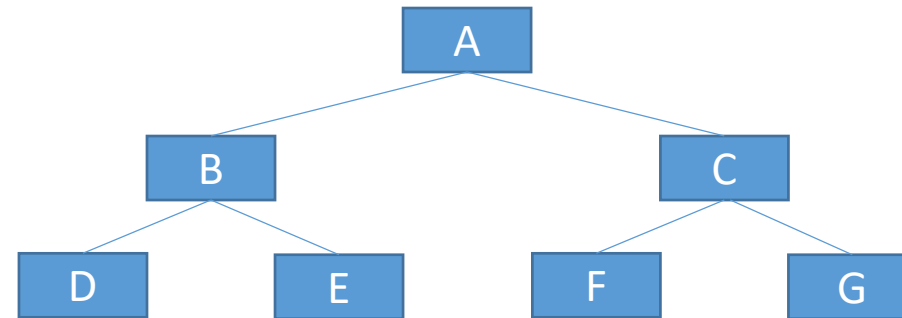
Visit_post (node)
Visit_post (left child)
Visit_post (right child)
Print (node)



- What output do we get if we list the above tree:
 - Using pre-order traversal?

Visit_pre (node)
Print (node)
Visit_pre (left child)
Visit_pre (right child)

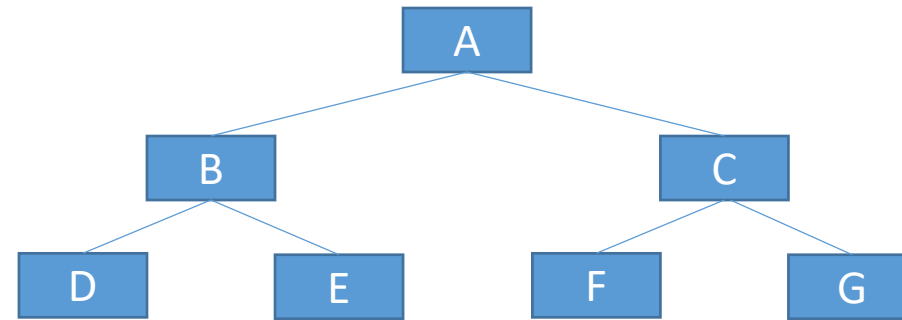
A, B, D, E, C , F, G



- What output do we get if we list the above tree:
 - Using in-order traversal?

Visit_in (node)
Visit_in (left child)
Print (node)
Visit_in (right child)

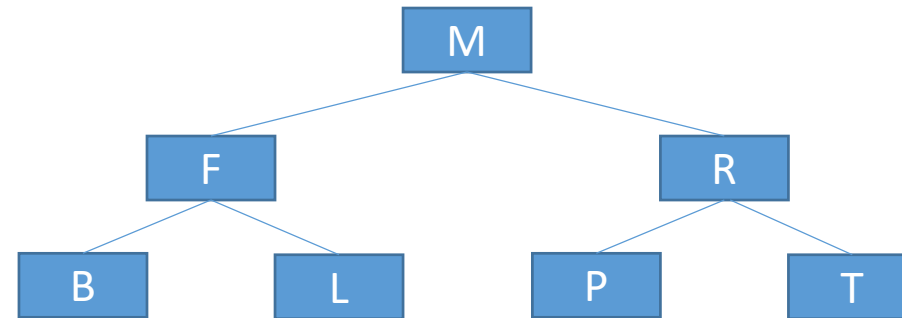
D, B, E, A, F, C, G



- What output do we get if we list the above tree:
 - Using post-order traversal?

Visit_post (node)
 Visit_post (left child)
 Visit_post (right child)
 Print (node)

D, E, B, F, G, C, A

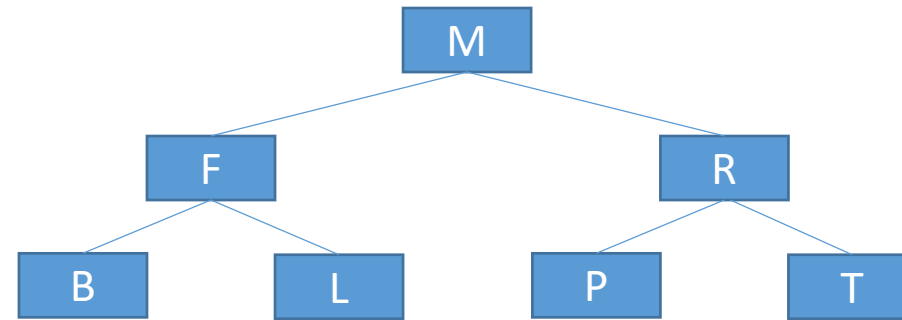


- If we traverse a binary search tree like the one above;
 - Which of the three traversal strategies makes the most sense?

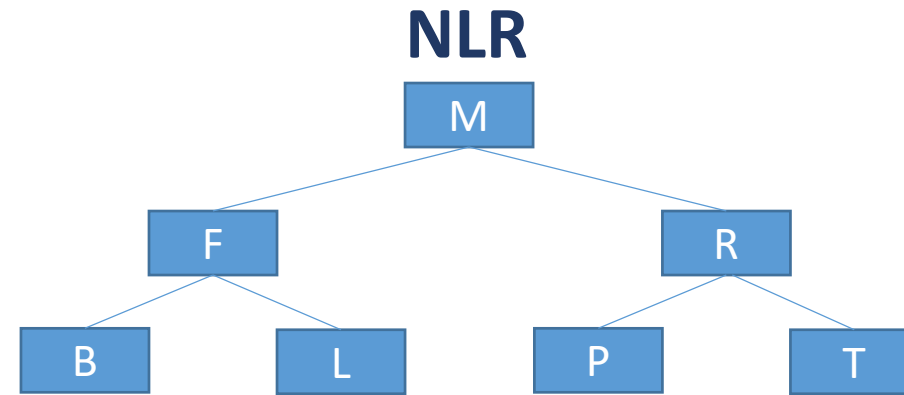
Visit_pre (node)
Print (node)
Visit_pre (left child)
Visit_pre (right child)

Visit_in (node)
Visit_in (left child)
Print (node)
Visit_in (right child)

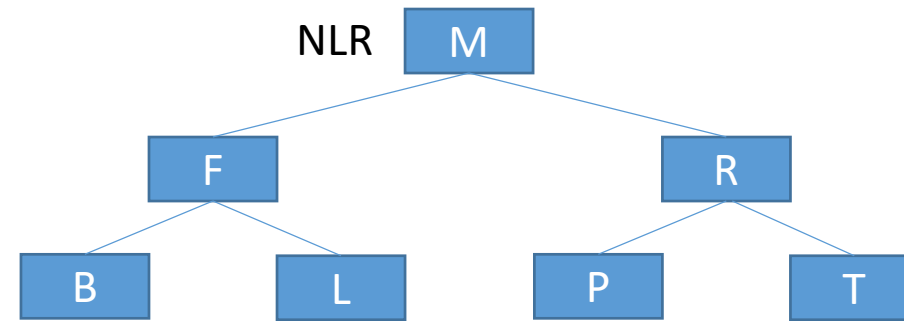
Visit_post (node)
Visit_post (left child)
Visit_post (right child)
Print (node)



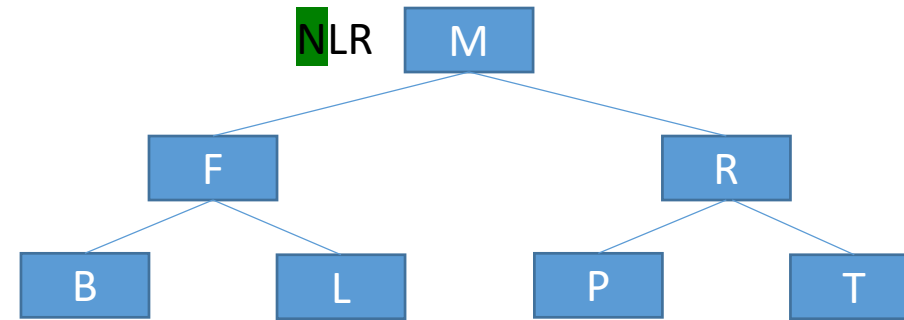
- Pre-order:
- In-order:
- Post-order:



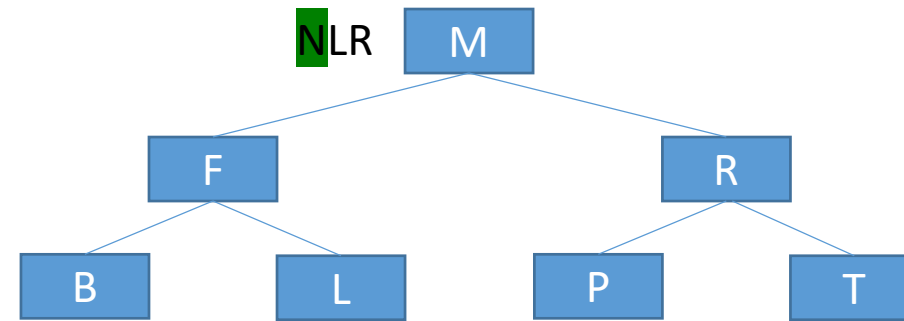
- Pre-order:
- In-order:
- Post-order:



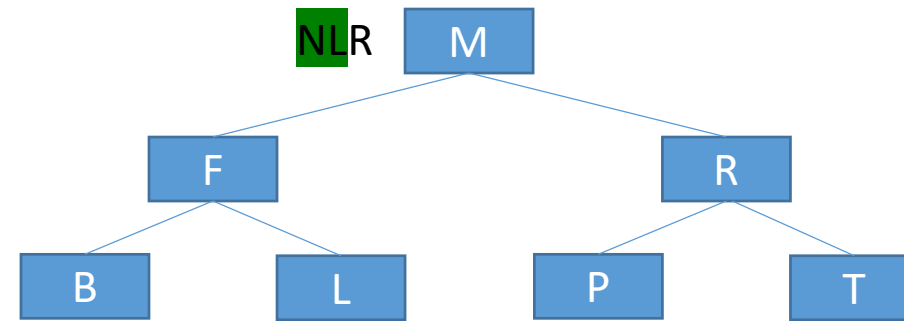
- Pre-order:
- In-order:
- Post-order:



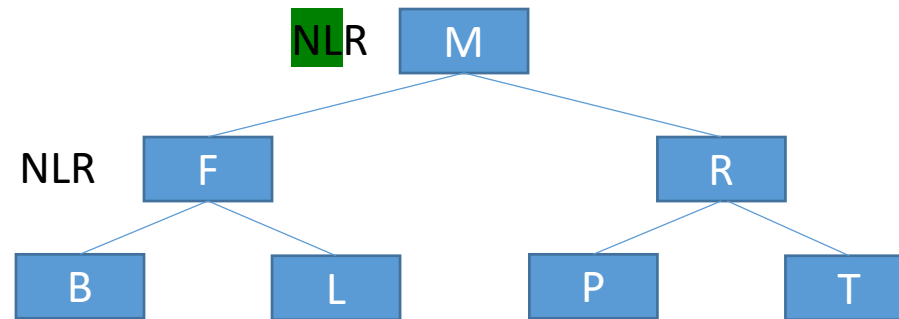
- Pre-order:
- In-order:
- Post-order:



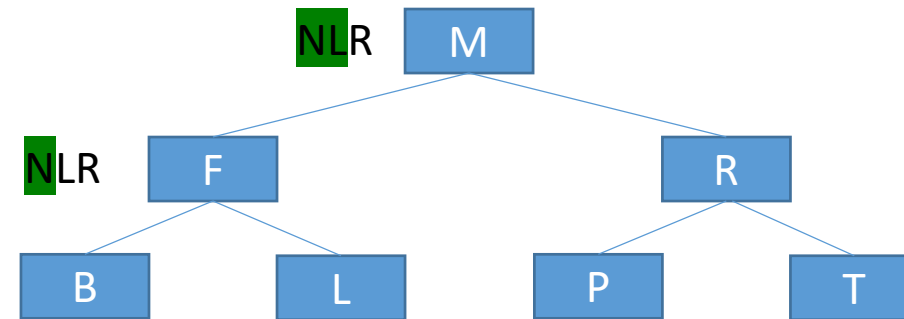
- Pre-order: M
- In-order:
- Post-order:



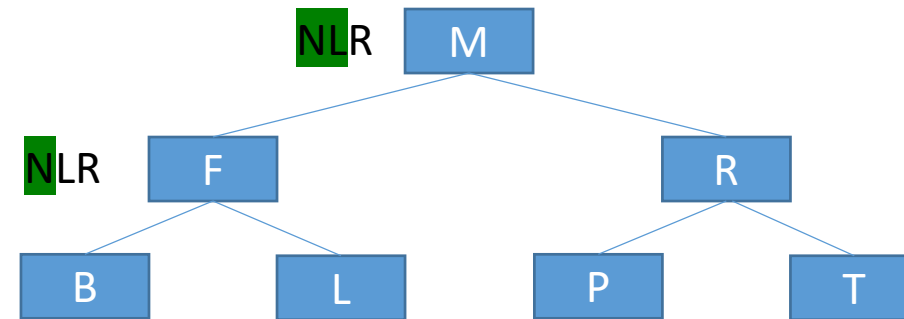
- Pre-order: M
- In-order:
- Post-order:



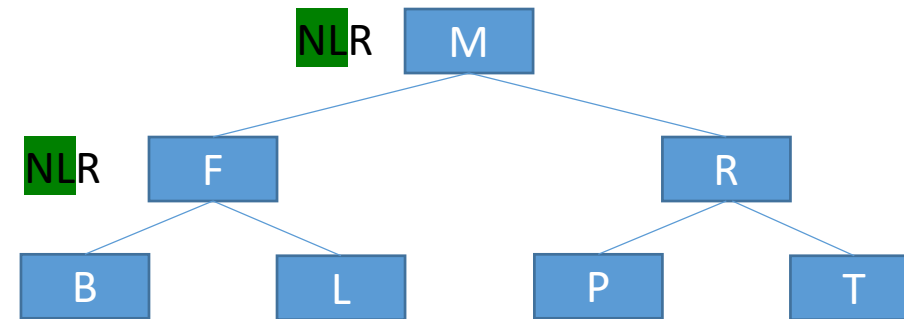
- Pre-order: M
- In-order:
- Post-order:



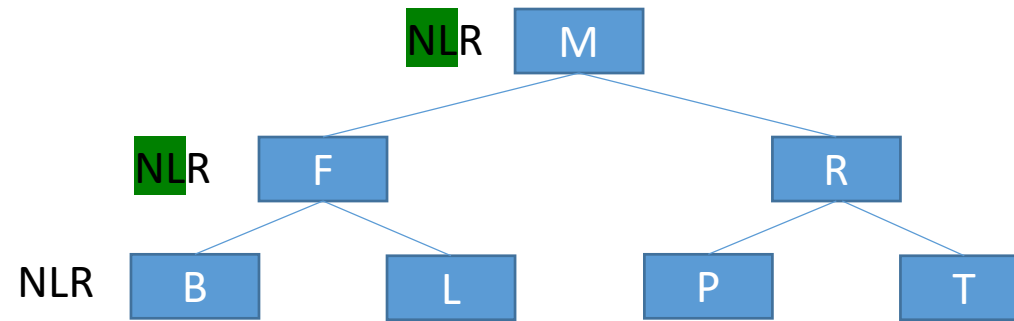
- Pre-order: M
- In-order:
- Post-order:



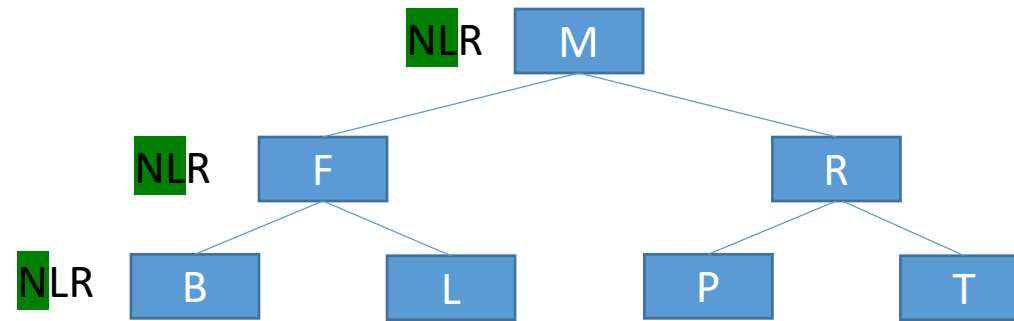
- Pre-order: M, F
- In-order:
- Post-order:



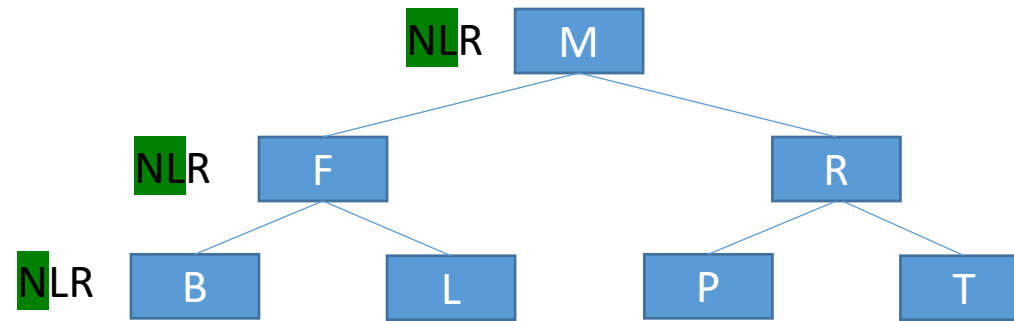
- Pre-order: M, F
- In-order:
- Post-order:



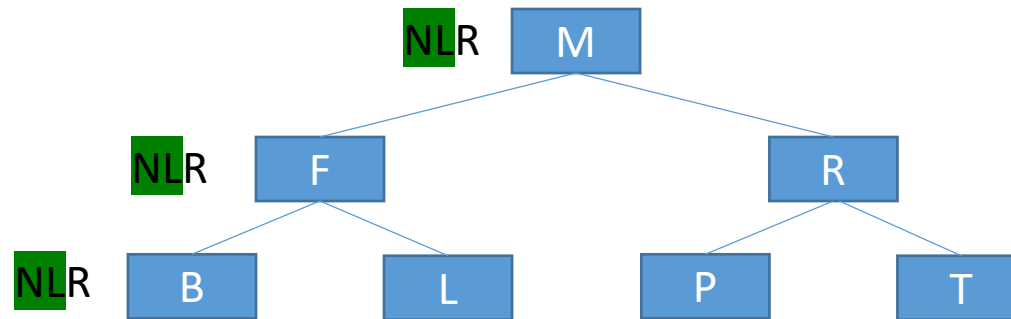
- Pre-order: M, F
- In-order:
- Post-order:



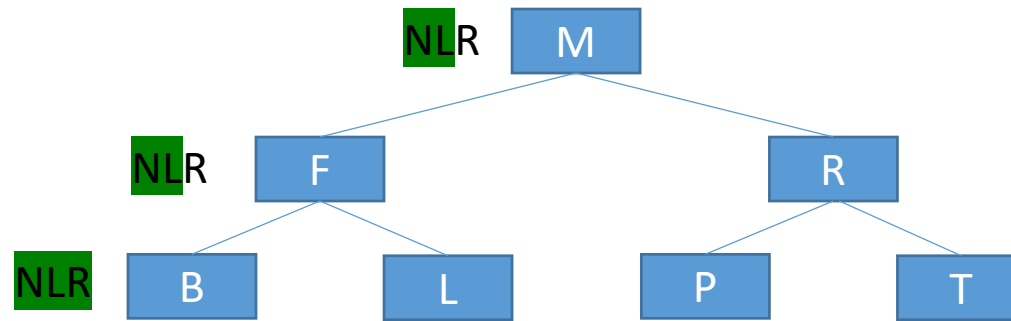
- Pre-order: M, F
- In-order:
- Post-order:



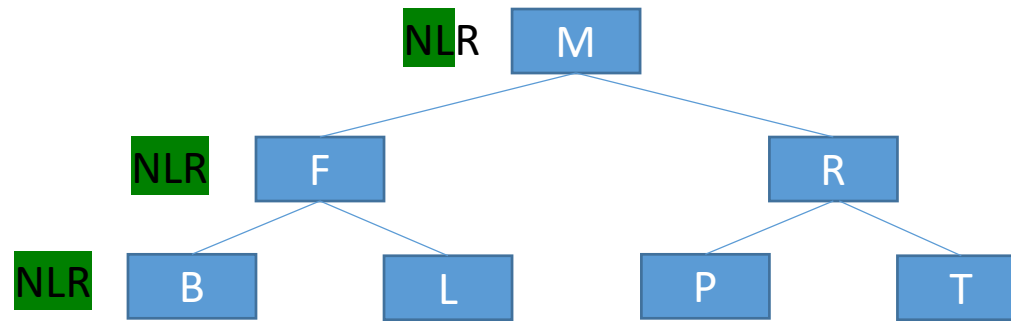
- Pre-order: M, F, B
- In-order:
- Post-order:



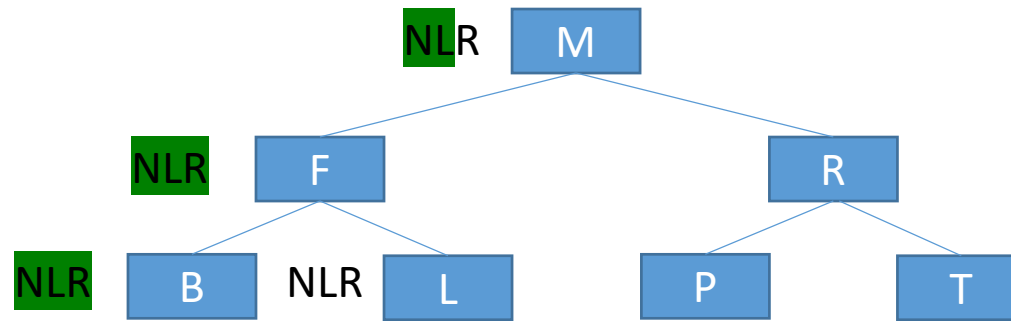
- Pre-order: M, F, B
- In-order:
- Post-order:



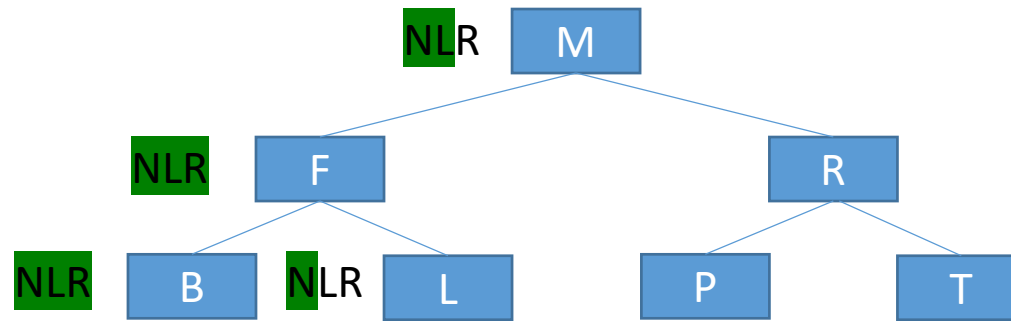
- Pre-order: M, F, B
- In-order:
- Post-order:



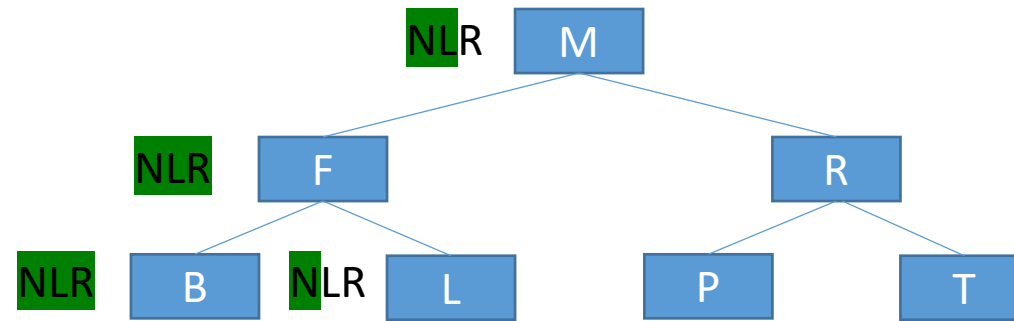
- Pre-order: M, F, B
- In-order:
- Post-order:



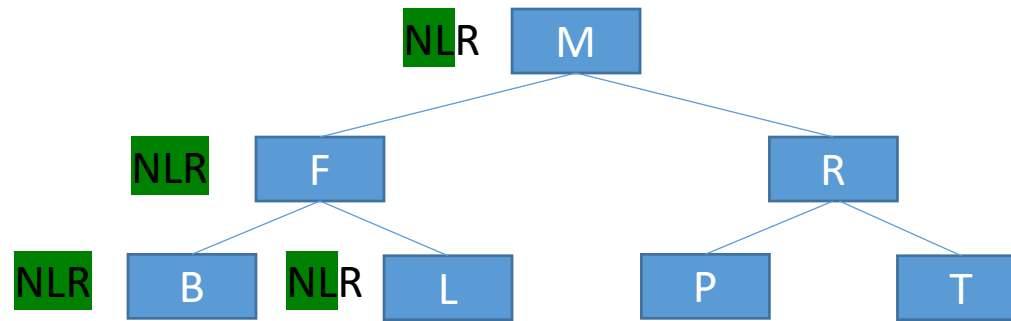
- Pre-order: M, F, B
- In-order:
- Post-order:



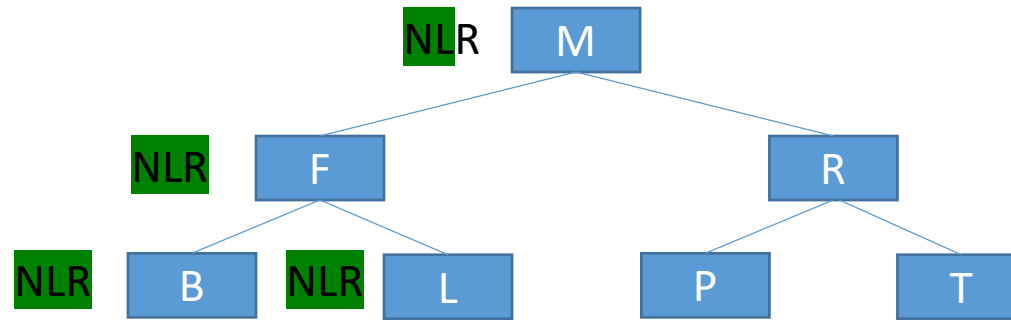
- Pre-order: M, F, B
- In-order:
- Post-order:



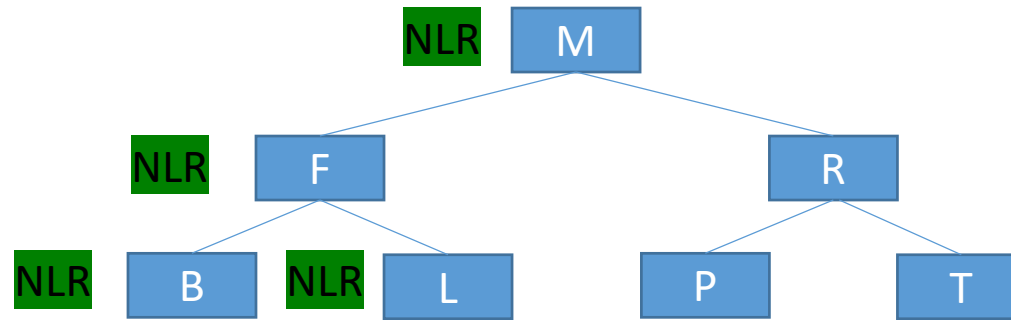
- Pre-order: M, F, B, L
- In-order:
- Post-order:



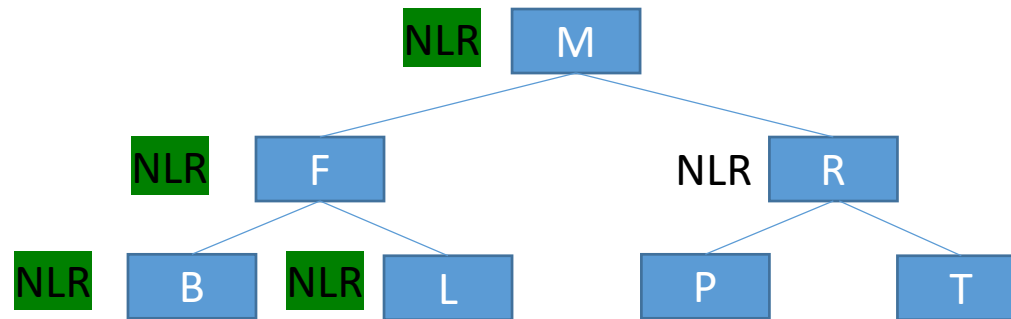
- Pre-order: M, F, B, L
- In-order:
- Post-order:



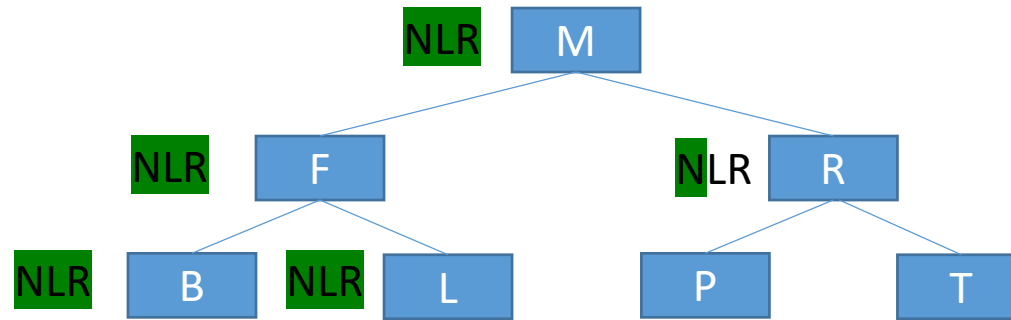
- Pre-order: M, F, B, L
- In-order:
- Post-order:



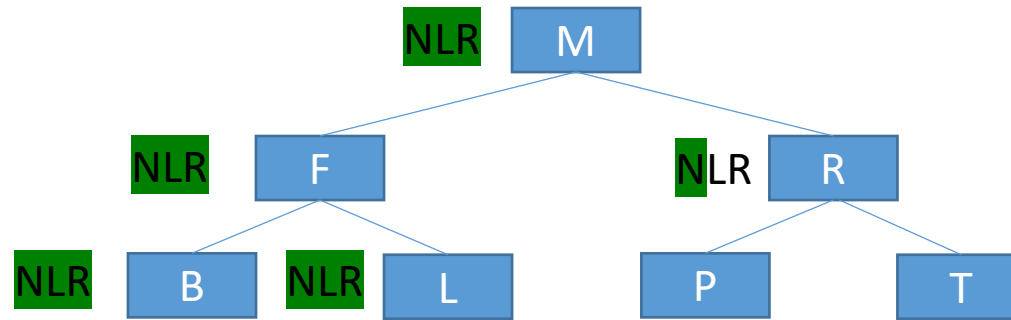
- Pre-order: M, F, B, L
- In-order:
- Post-order:



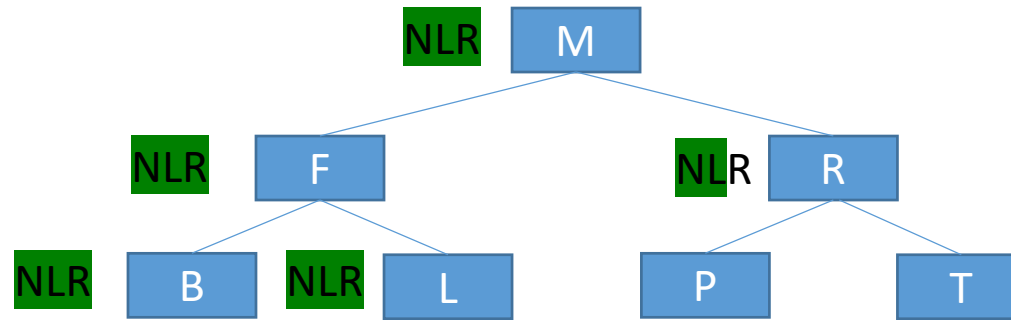
- Pre-order: M, F, B, L
- In-order:
- Post-order:



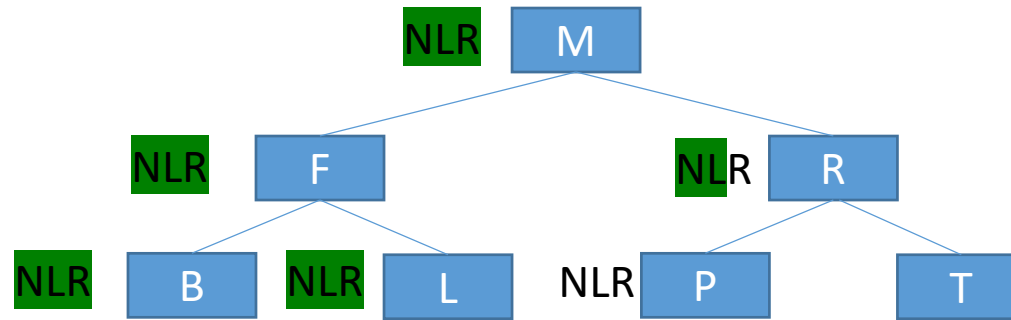
- Pre-order: M, F, B, L
- In-order:
- Post-order:



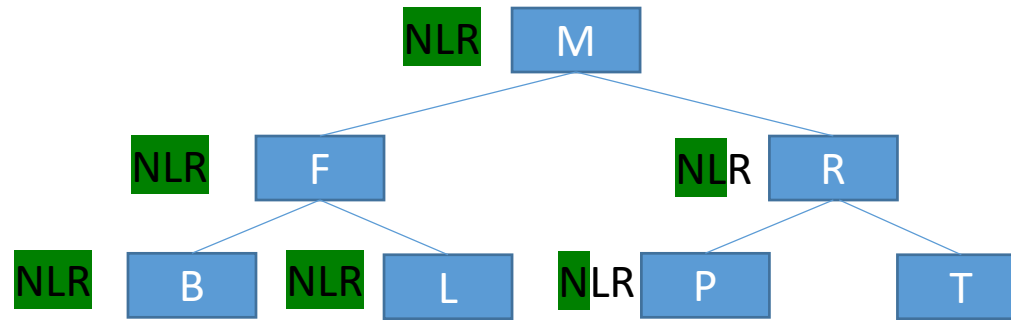
- Pre-order: M, F, B, L, R
- In-order:
- Post-order:



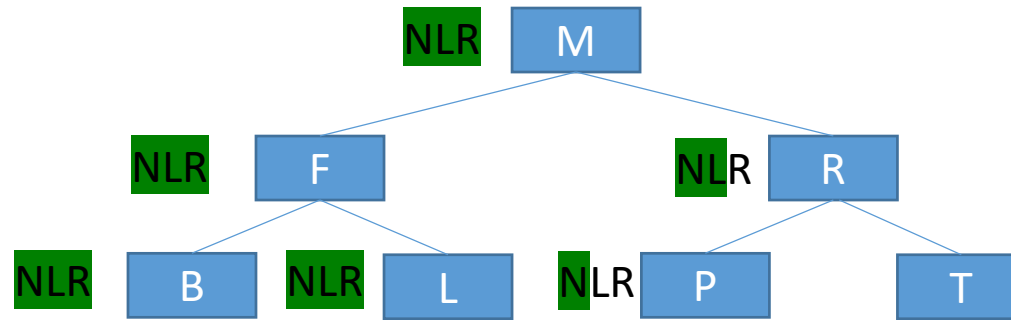
- Pre-order: M, F, B, L, R
- In-order:
- Post-order:



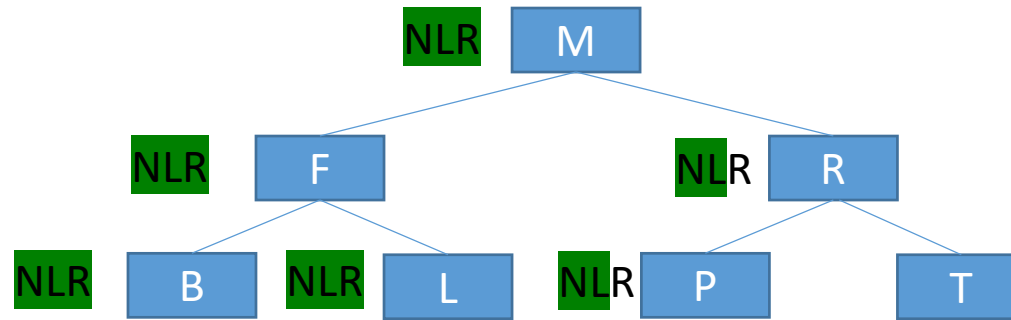
- Pre-order: M, F, B, L, R
- In-order:
- Post-order:



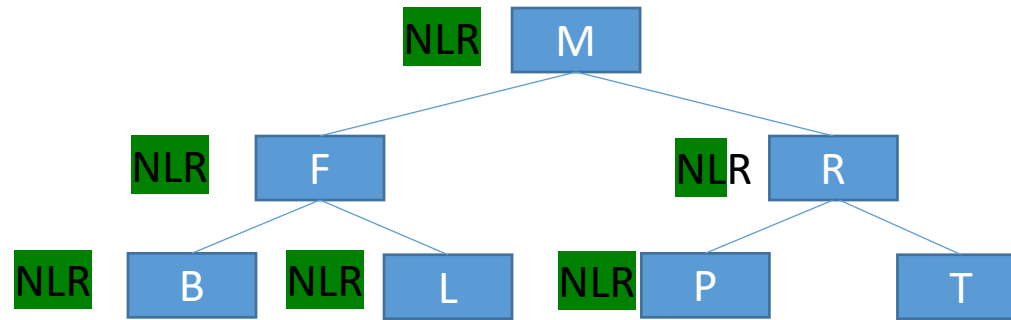
- Pre-order: M, F, B, L, R
- In-order:
- Post-order:



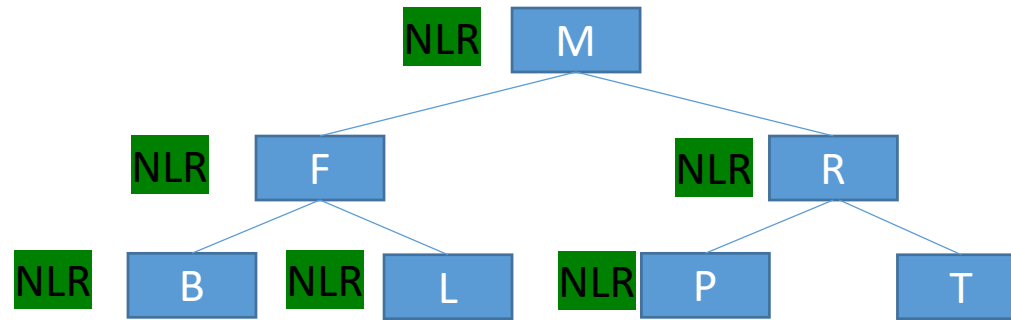
- Pre-order: M, F, B, L, R, P
- In-order:
- Post-order:



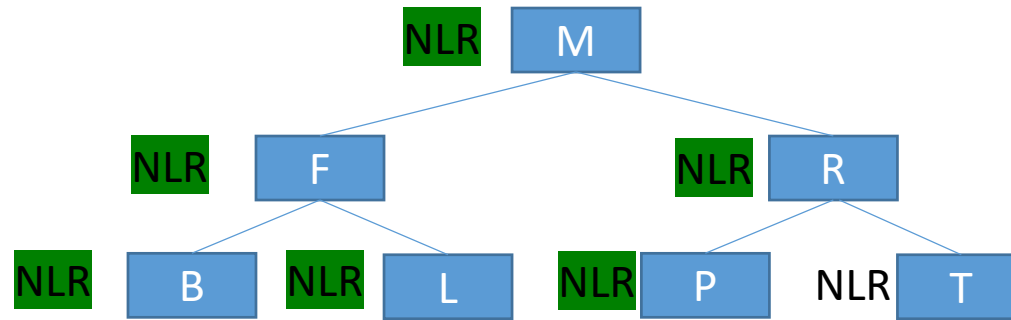
- Pre-order: M, F, B, L, R, P
- In-order:
- Post-order:



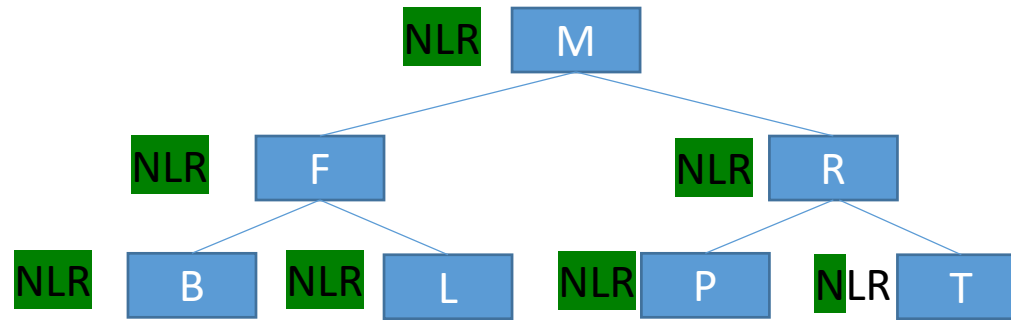
- Pre-order: M, F, B, L, R, P
- In-order:
- Post-order:



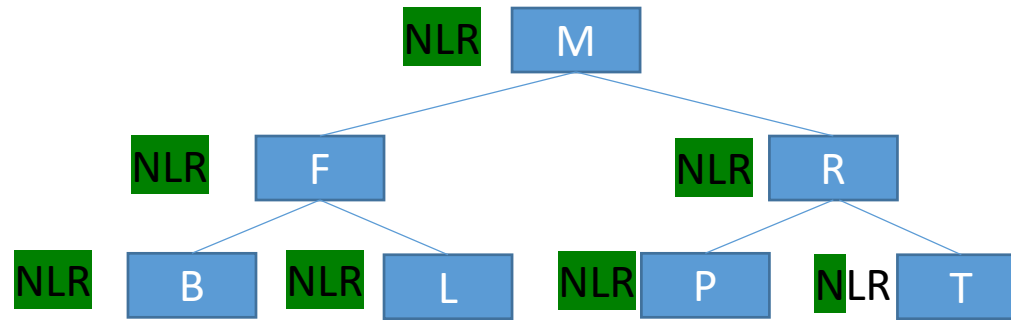
- Pre-order: M, F, B, L, R, P
- In-order:
- Post-order:



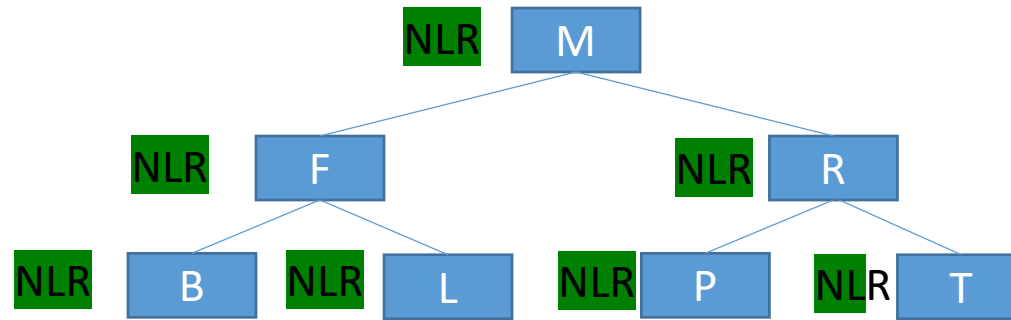
- Pre-order: M, F, B, L, R, P
- In-order:
- Post-order:



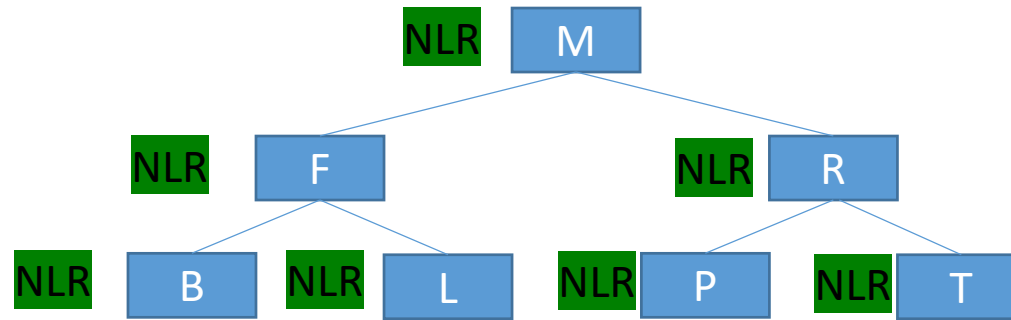
- Pre-order: M, F, B, L, R, P
- In-order:
- Post-order:



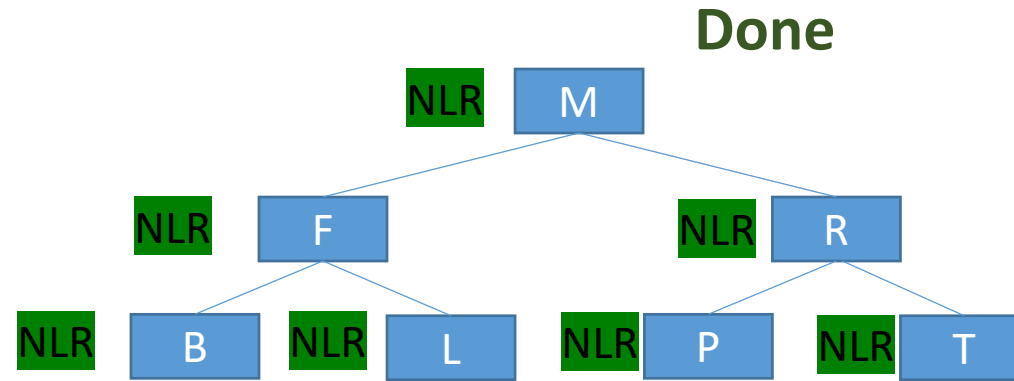
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



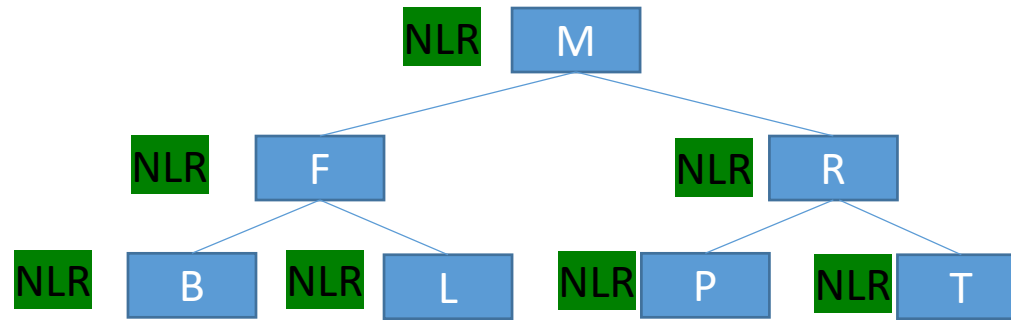
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



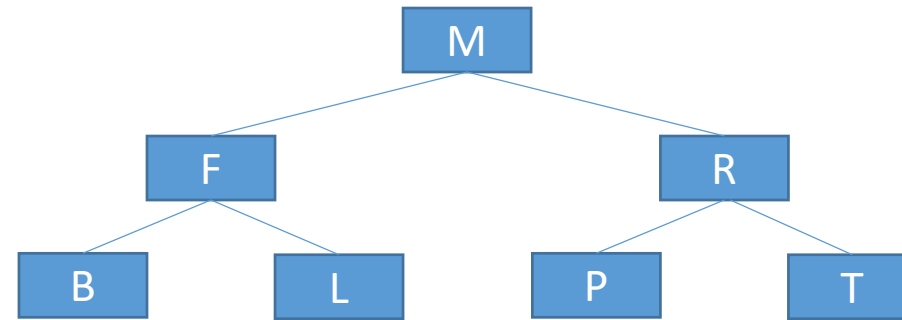
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



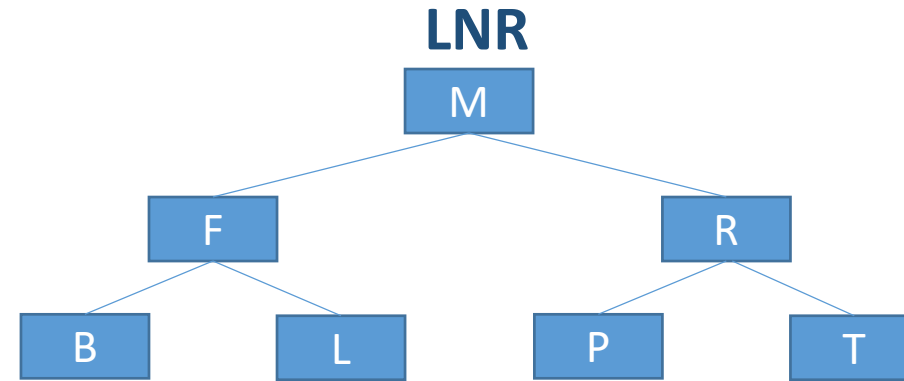
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



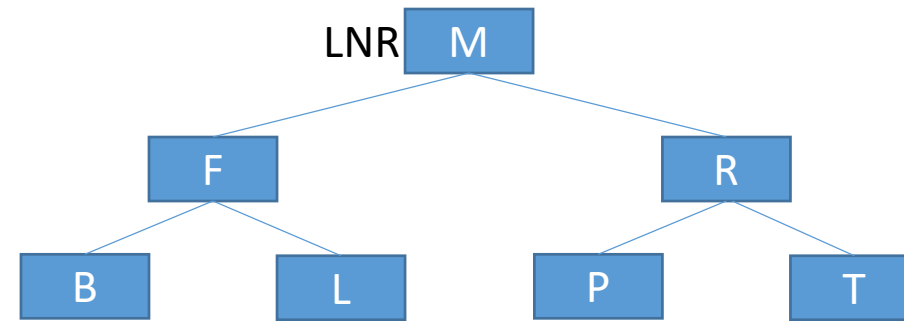
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



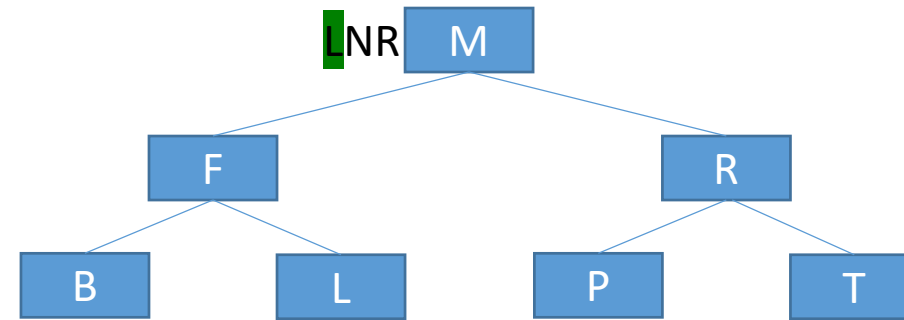
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



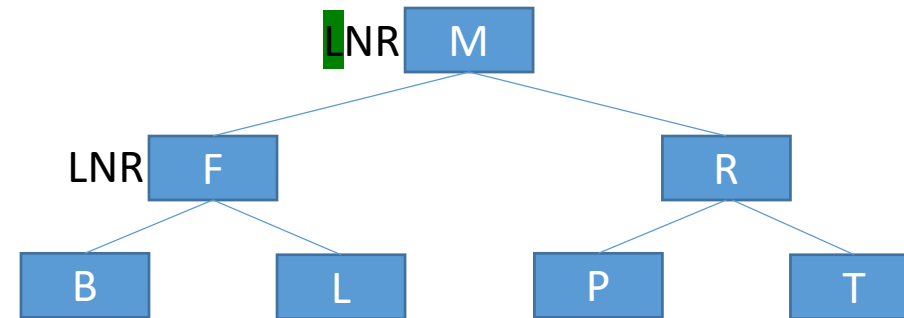
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



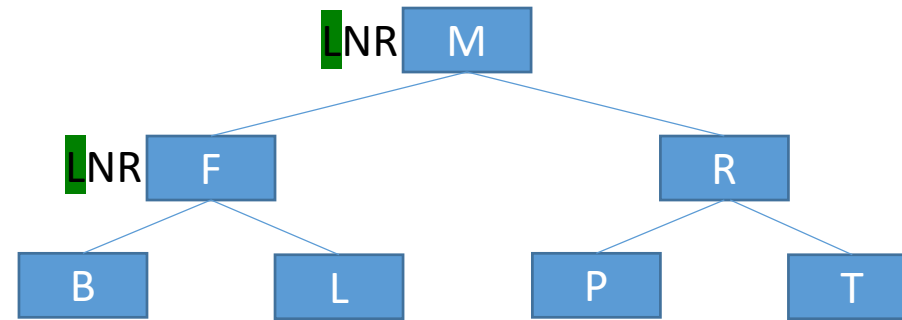
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



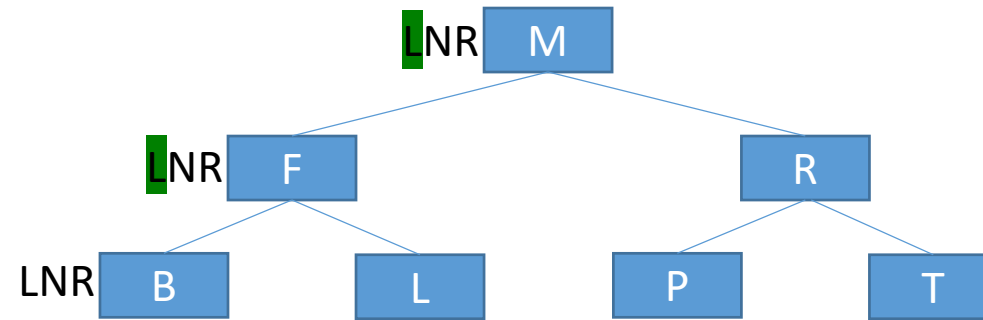
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



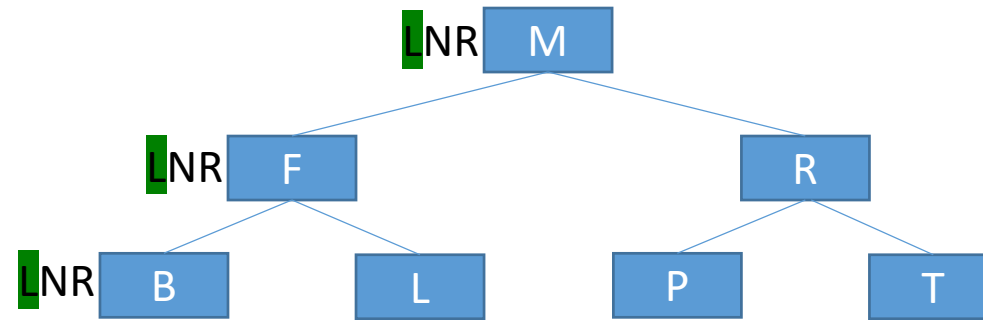
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



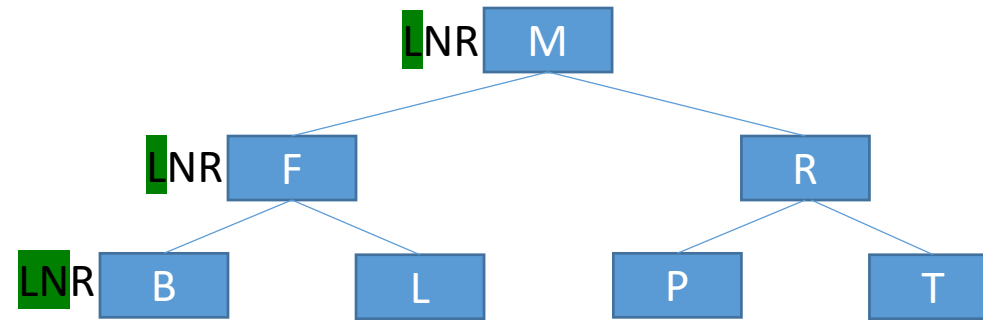
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



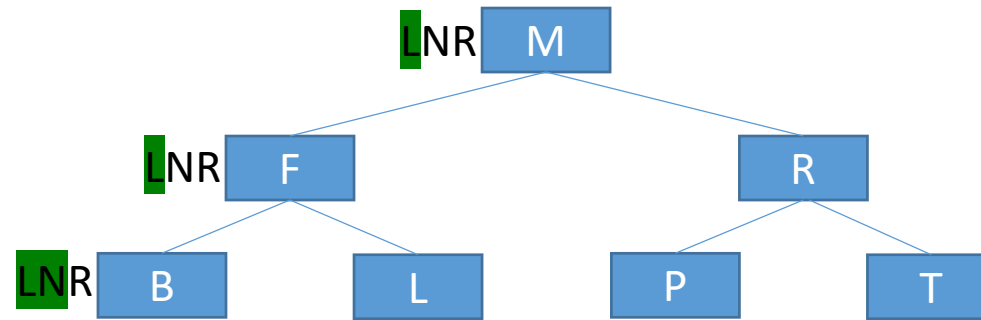
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



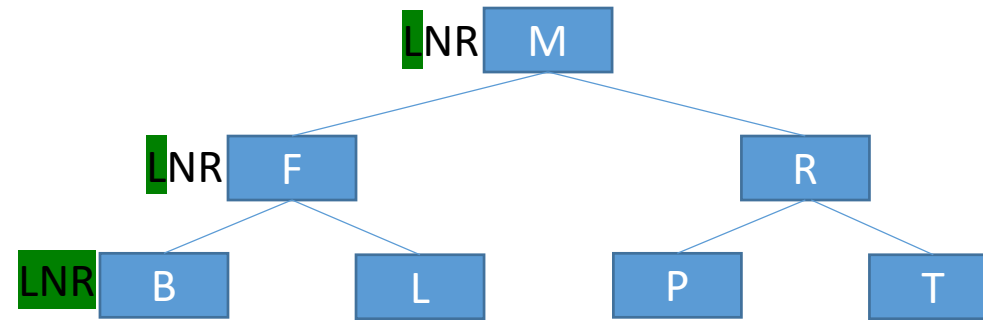
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



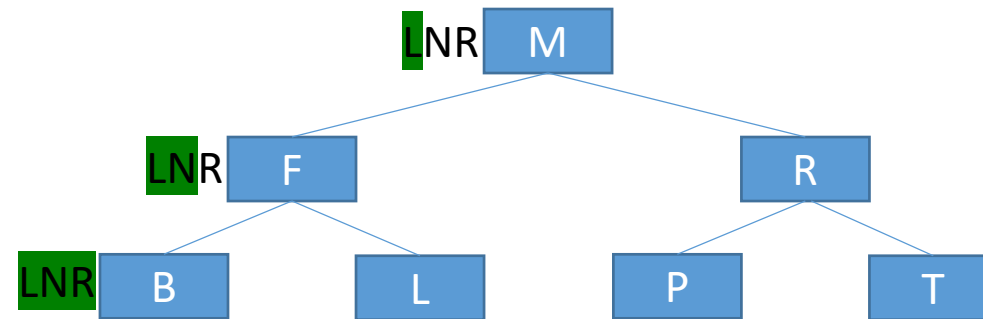
- Pre-order: M, F, B, L, R, P, T
- In-order:
- Post-order:



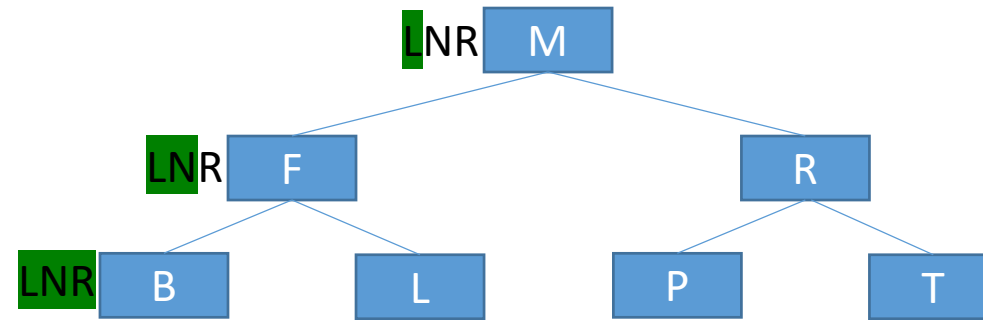
- Pre-order: M, F, B, L, R, P, T
- In-order: B
- Post-order:



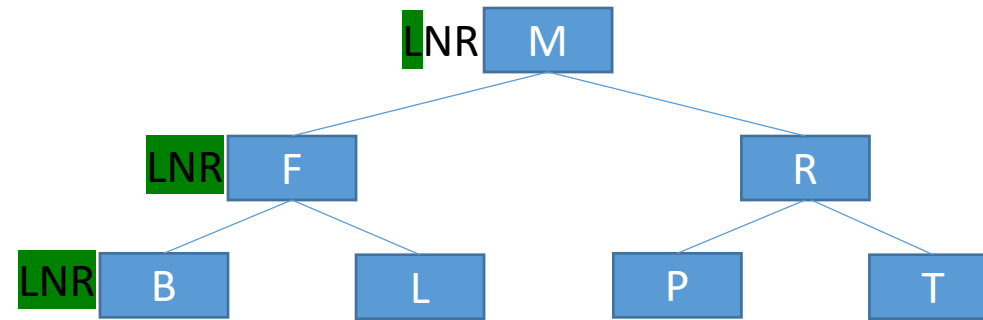
- Pre-order: M, F, B, L, R, P, T
- In-order: B
- Post-order:



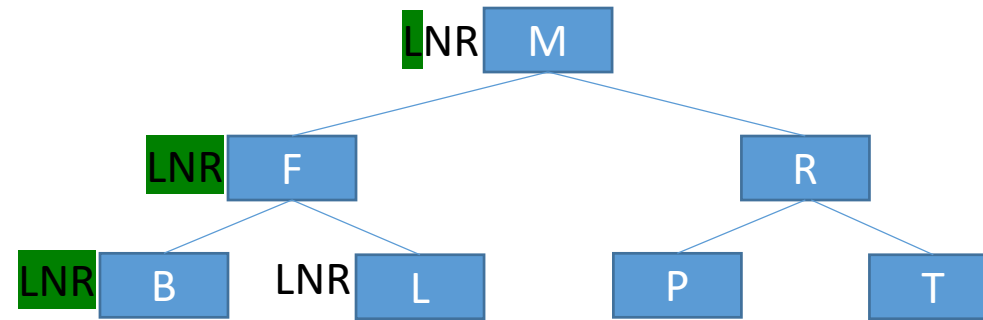
- Pre-order: M, F, B, L, R, P, T
- In-order: B
- Post-order:



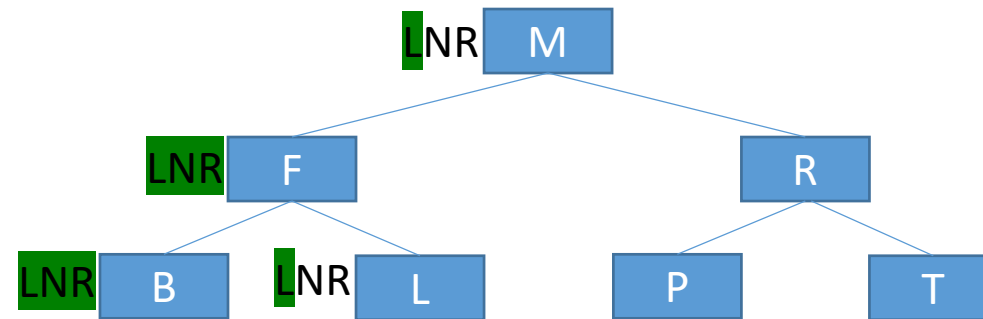
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F
- Post-order:



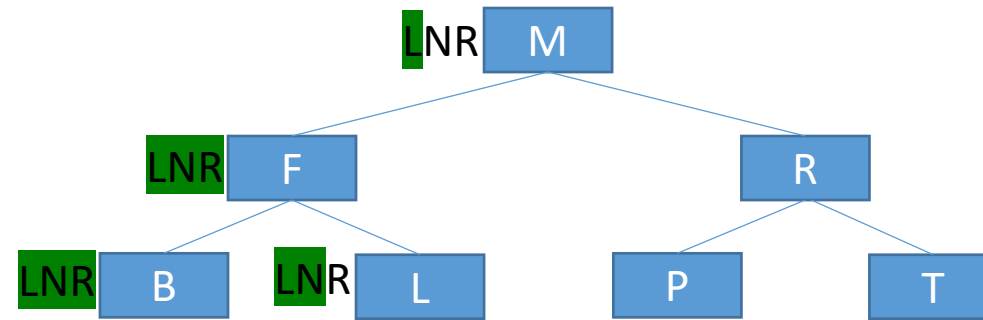
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F
- Post-order:



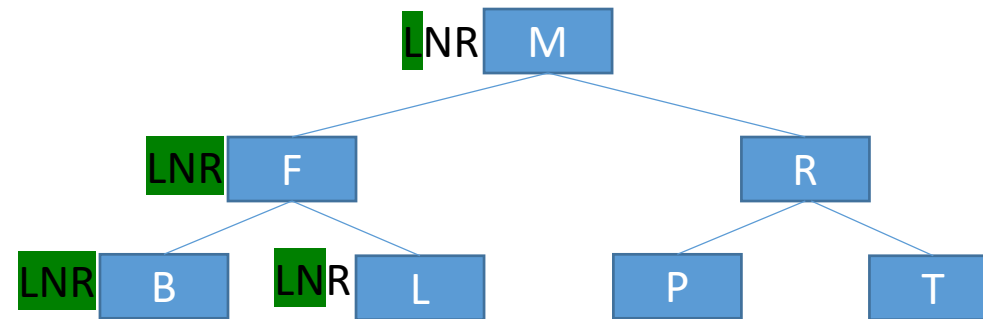
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F
- Post-order:



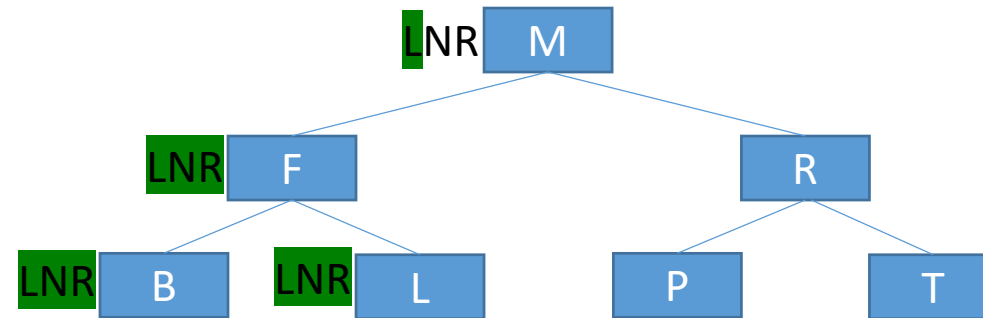
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F
- Post-order:



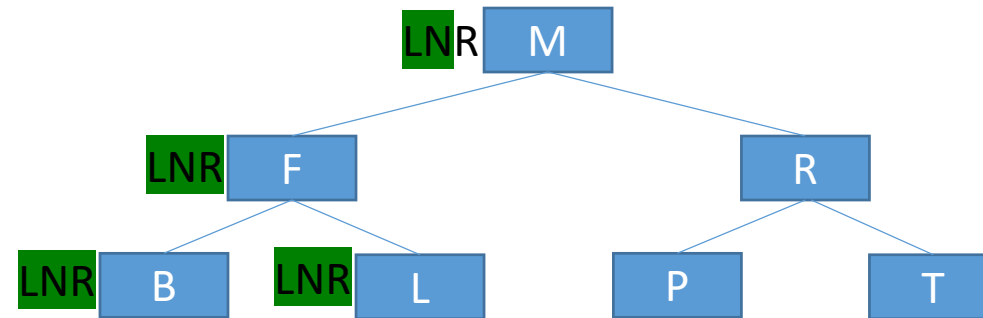
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F
- Post-order:



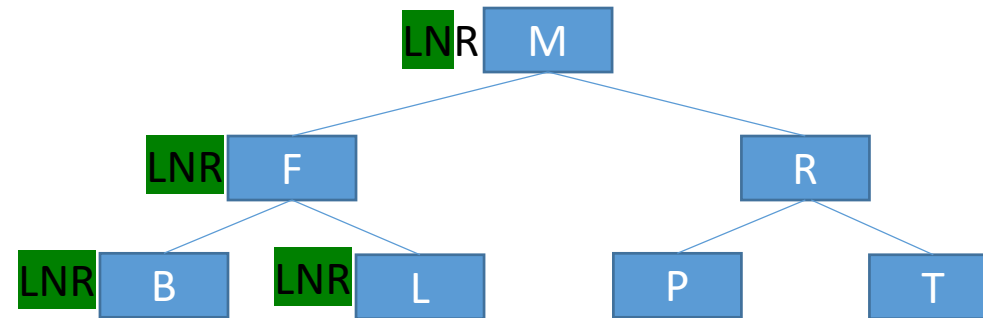
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L
- Post-order:



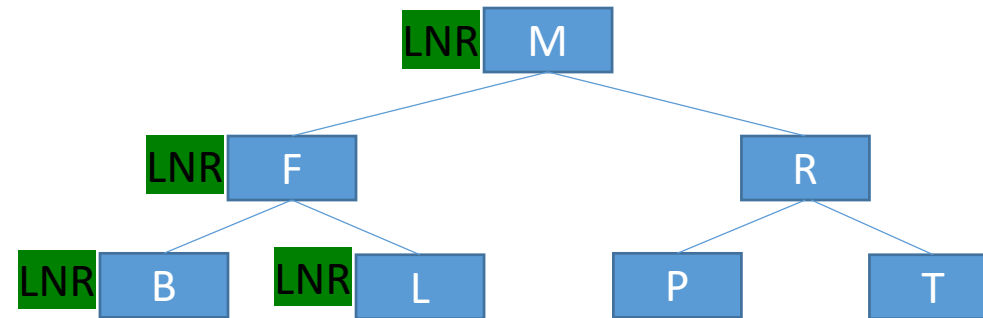
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L
- Post-order:



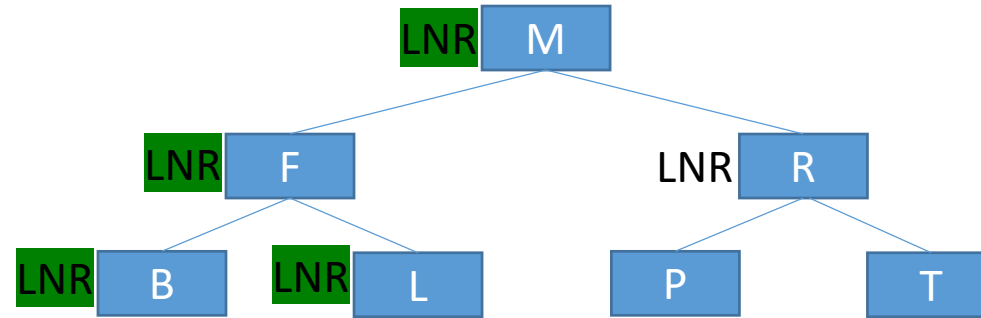
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L
- Post-order:



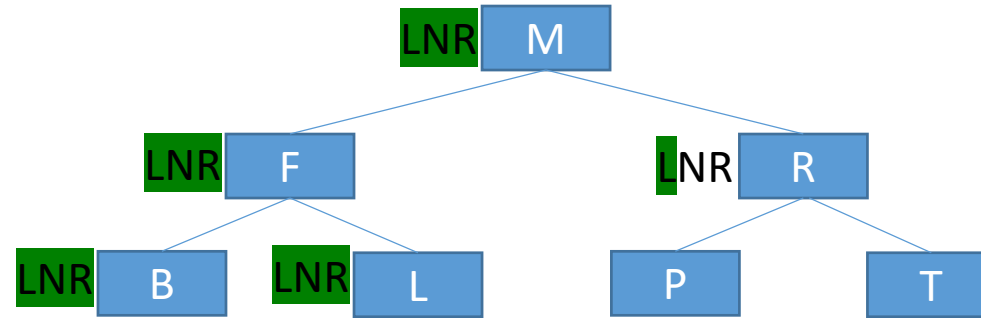
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



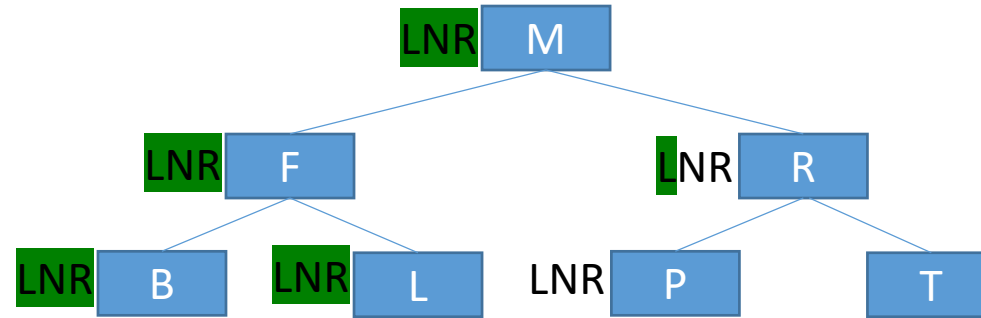
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



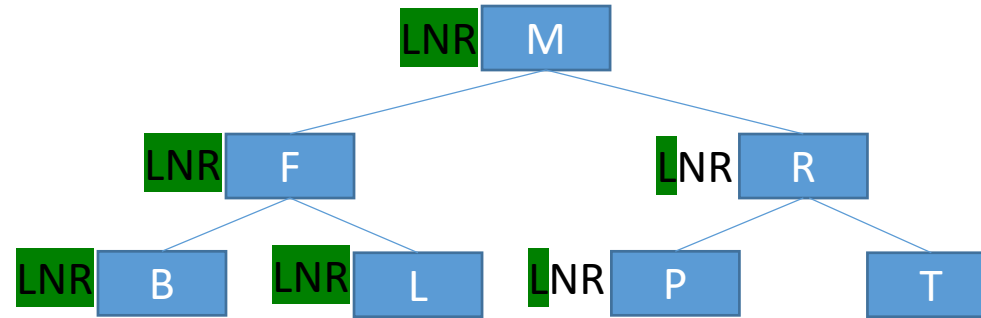
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



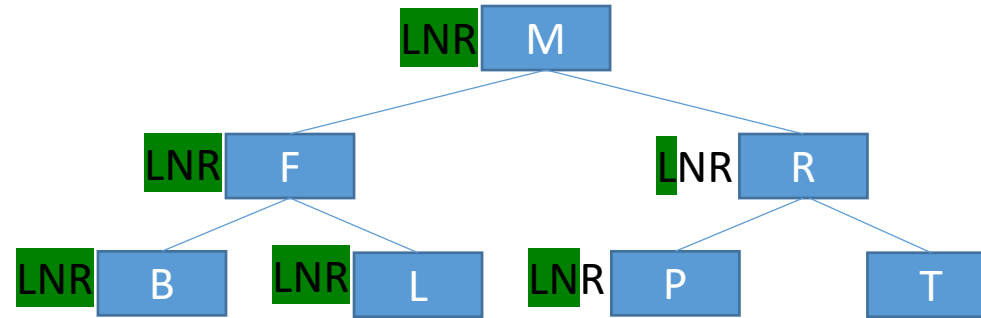
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



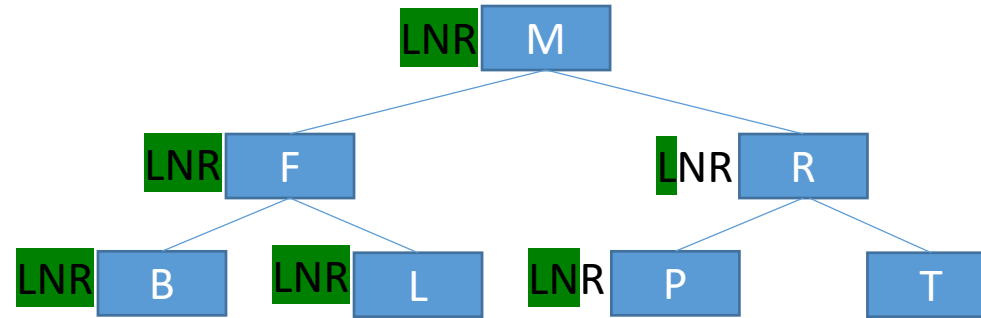
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



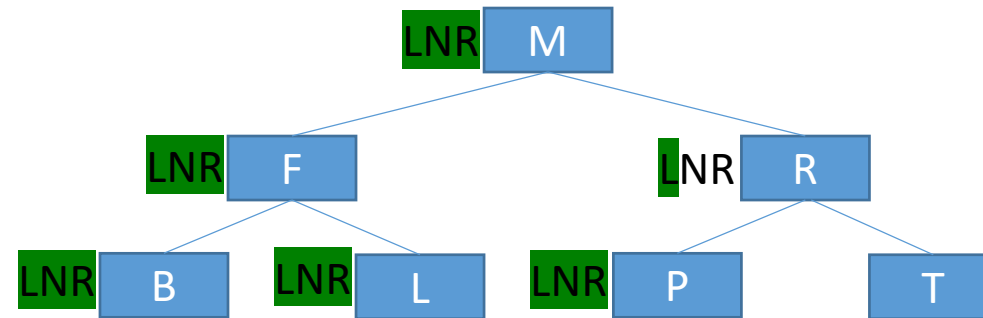
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



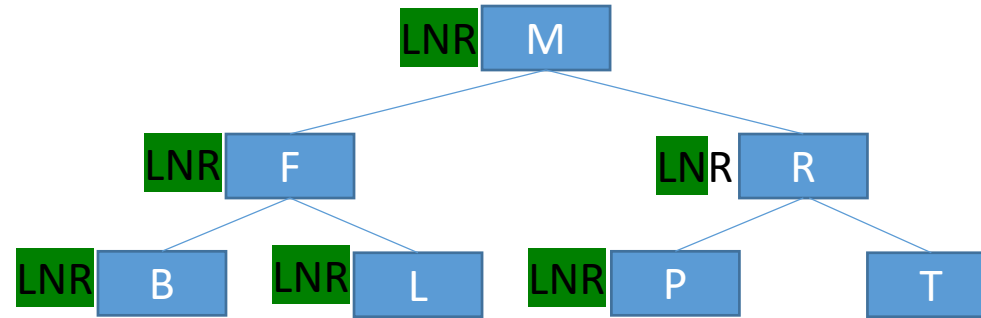
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M
- Post-order:



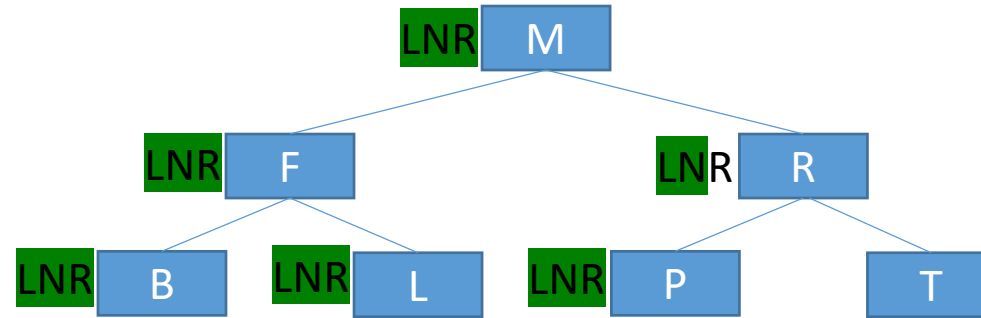
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P
- Post-order:



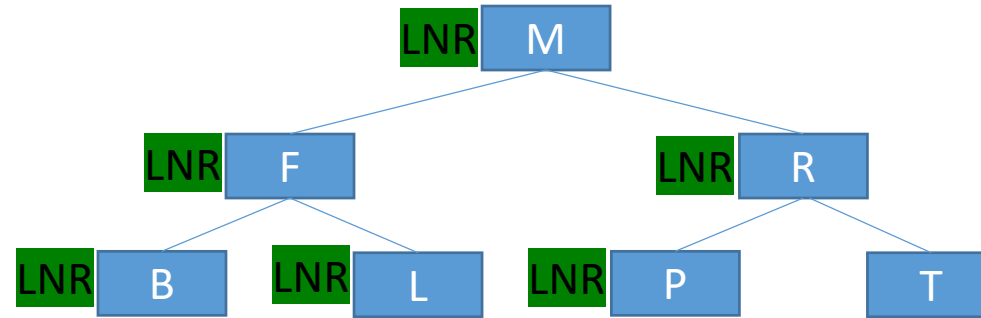
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P
- Post-order:



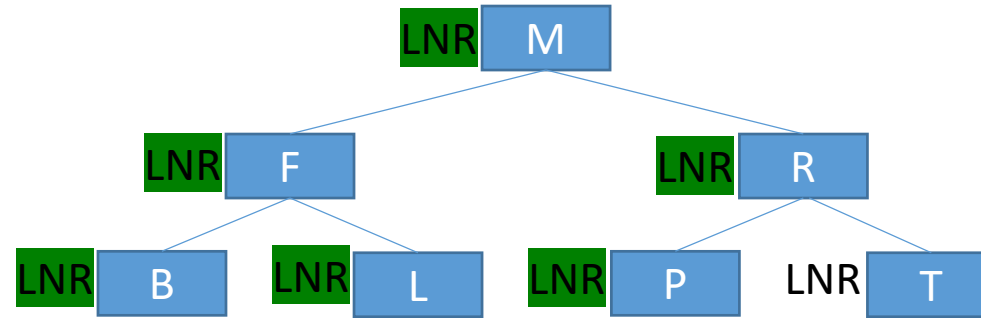
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P
- Post-order:



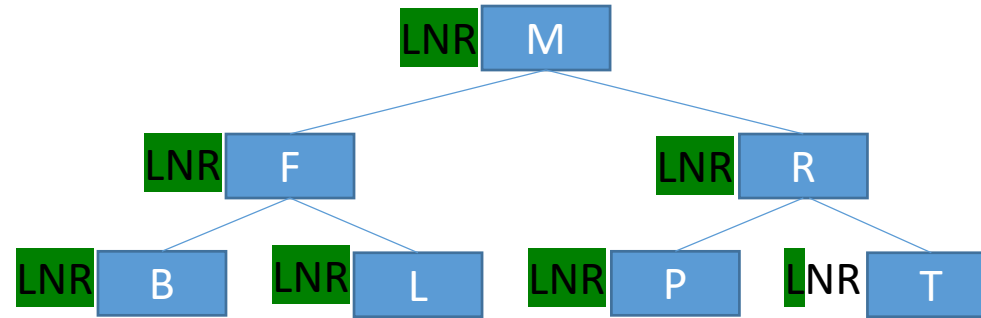
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R
- Post-order:



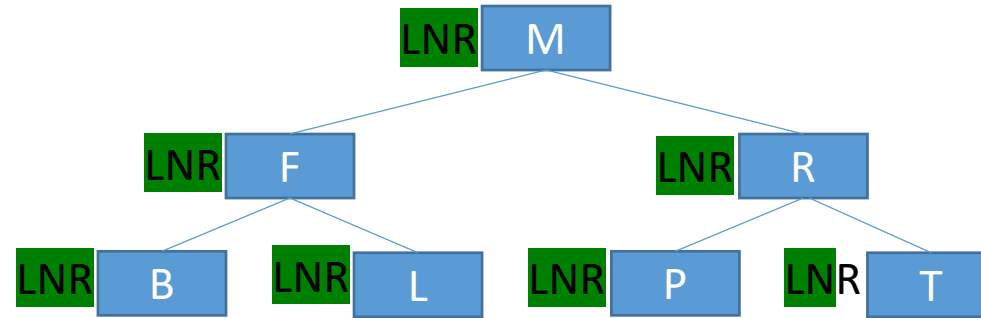
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R
- Post-order:



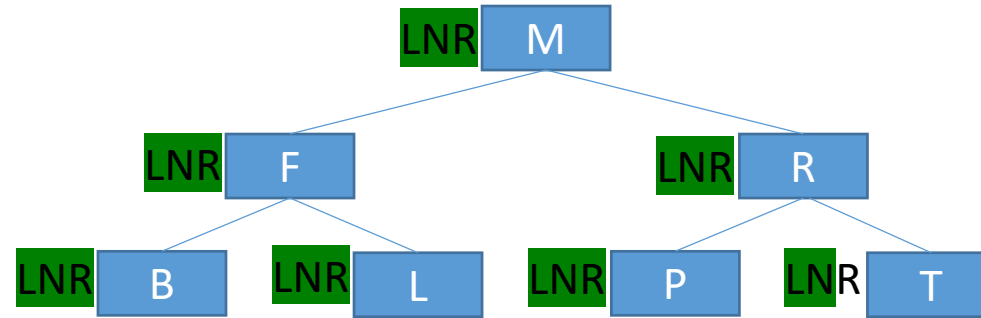
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R
- Post-order:



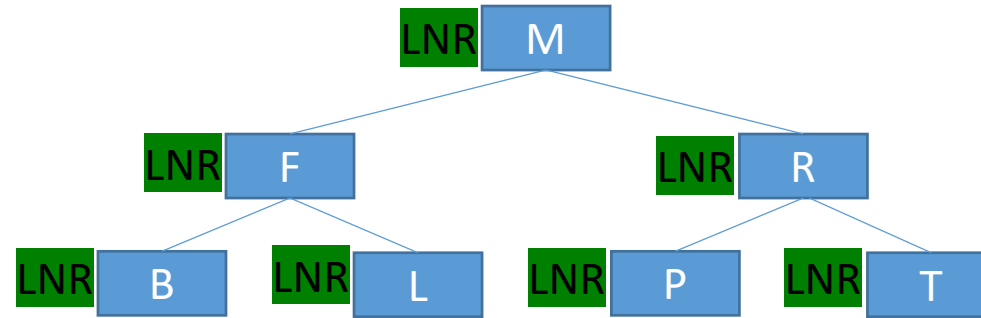
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R
- Post-order:



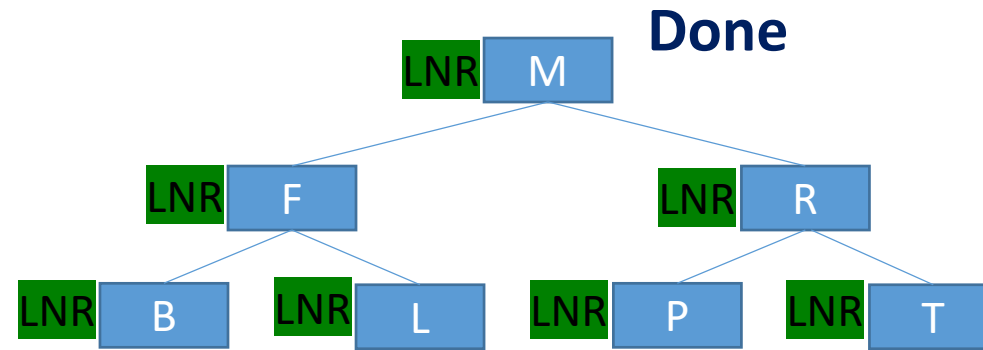
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R
- Post-order:



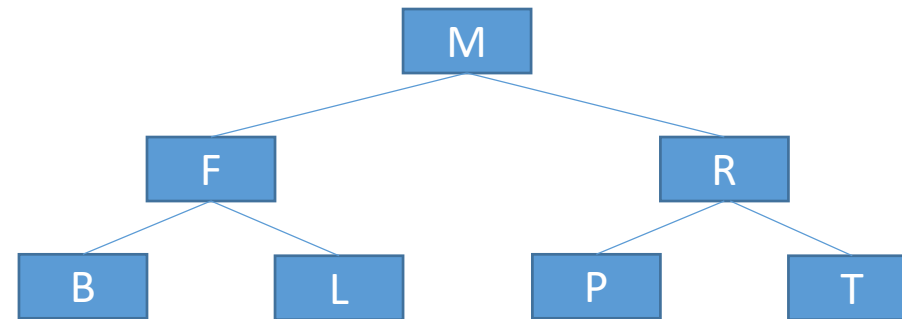
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



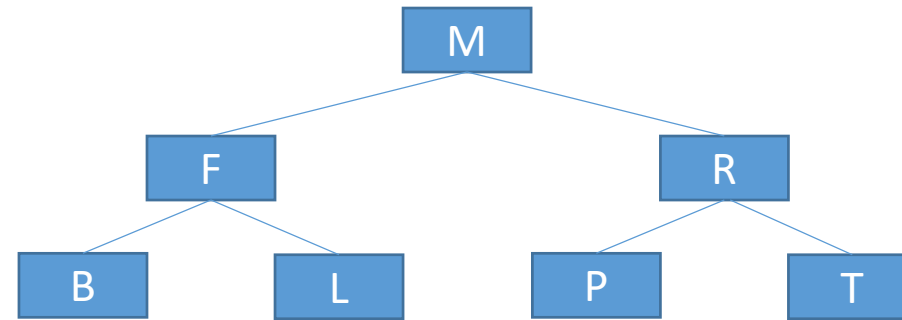
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



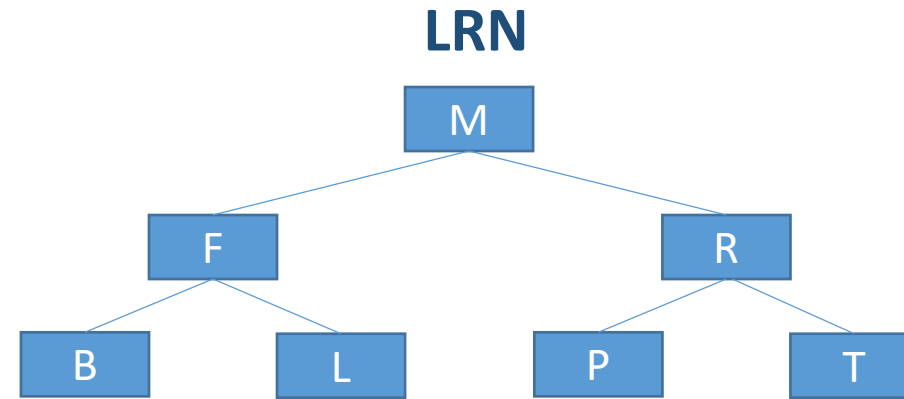
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



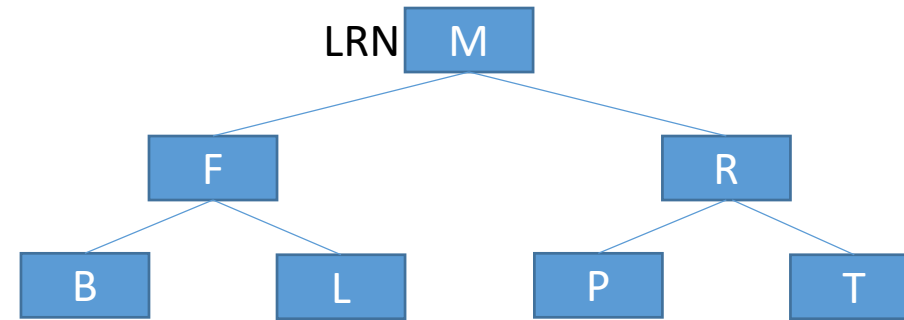
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



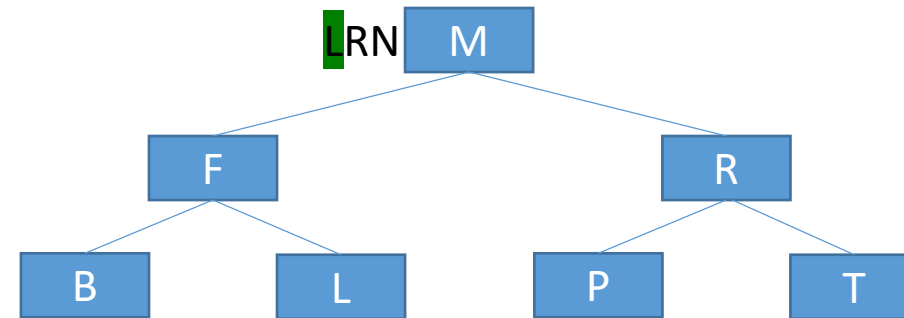
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



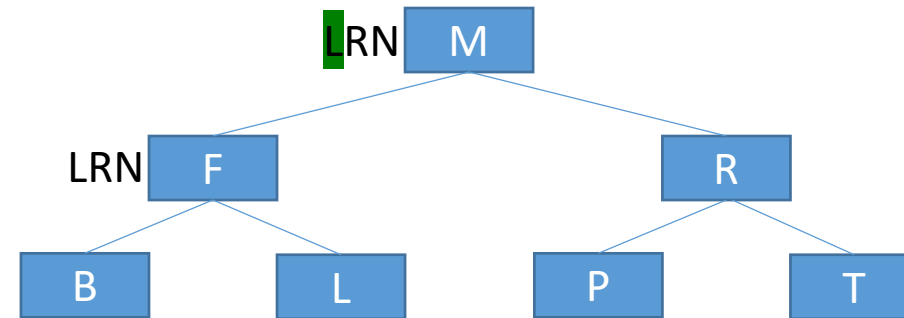
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



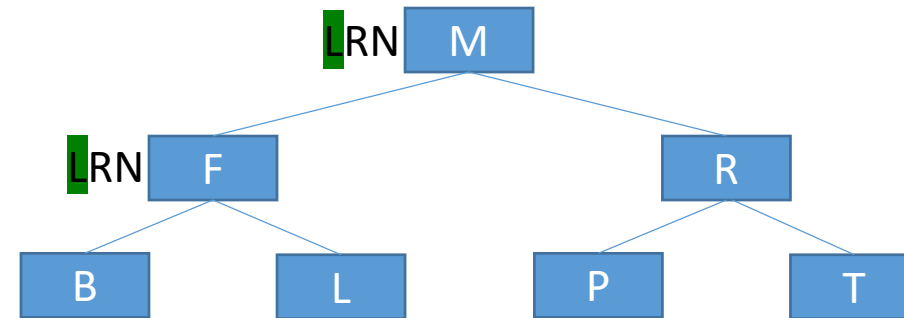
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



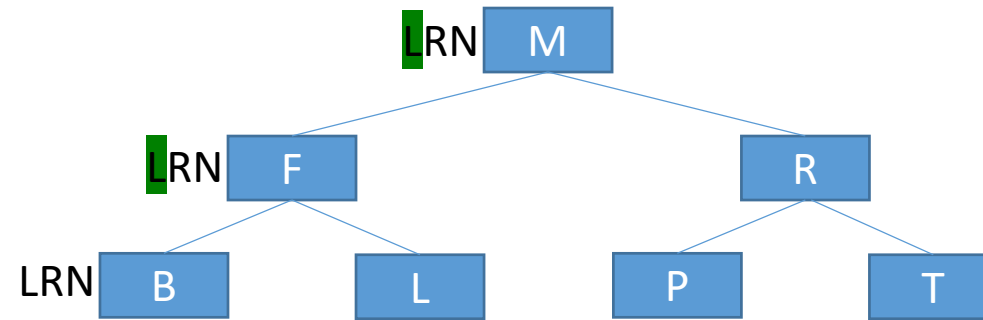
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



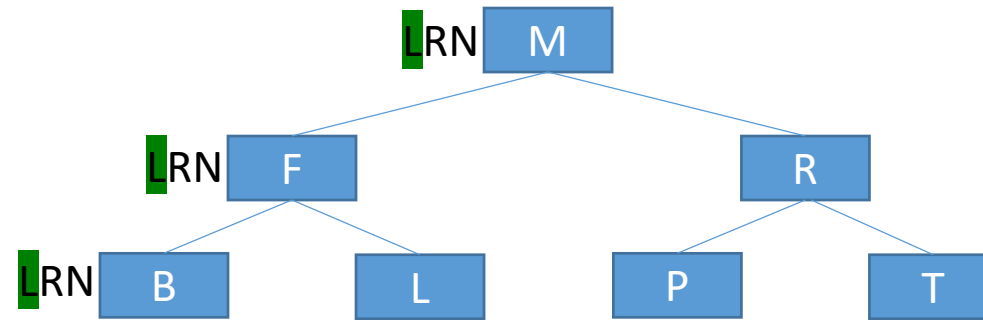
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



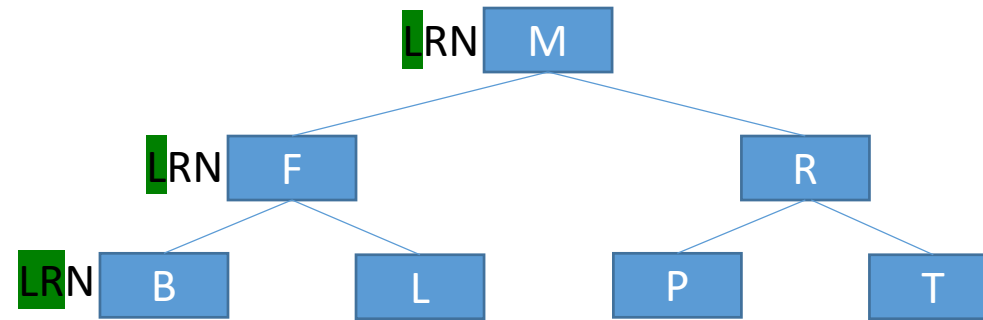
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



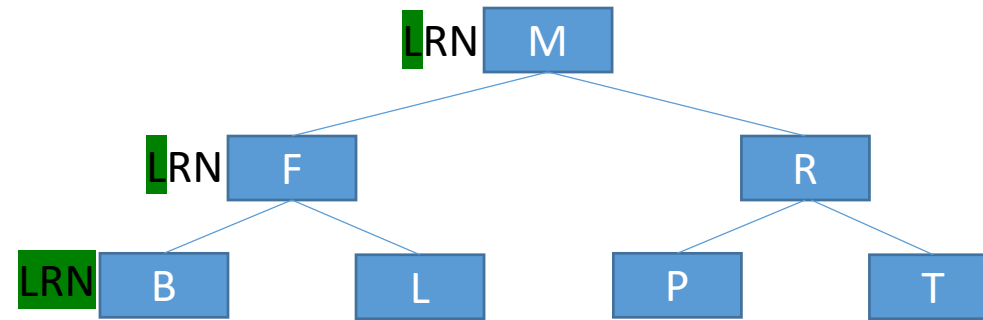
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



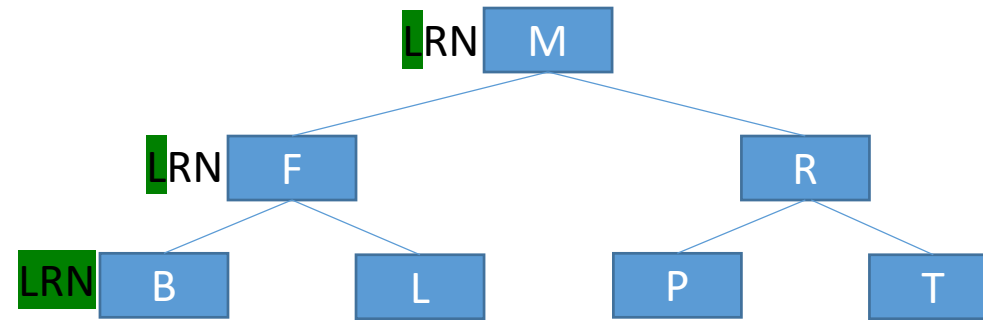
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



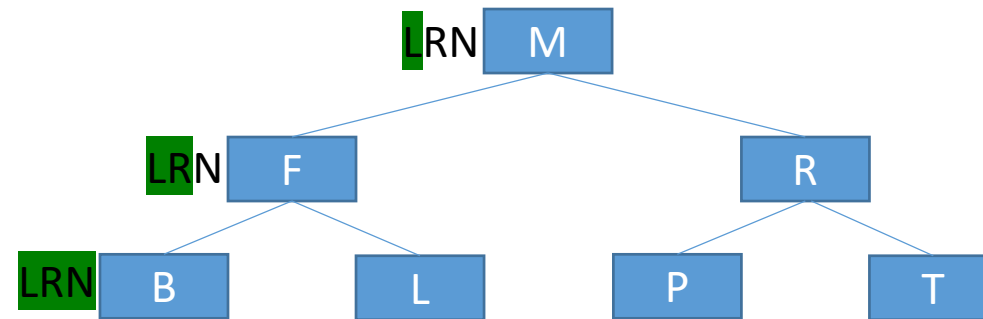
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



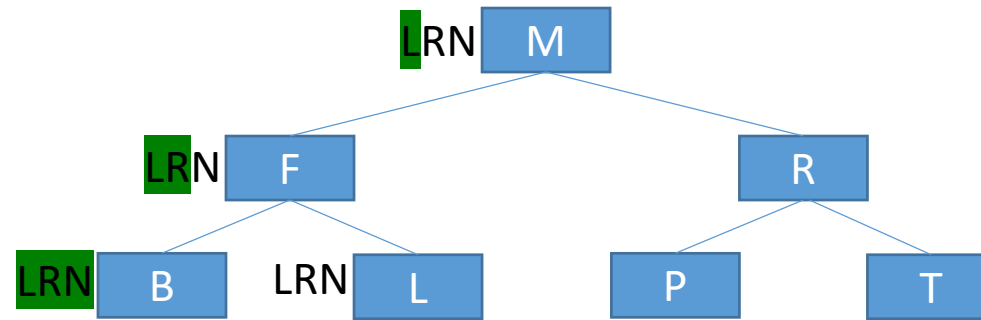
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order:



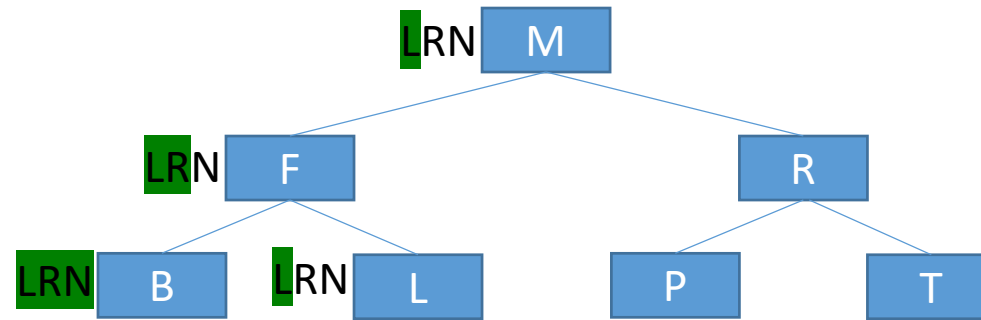
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B



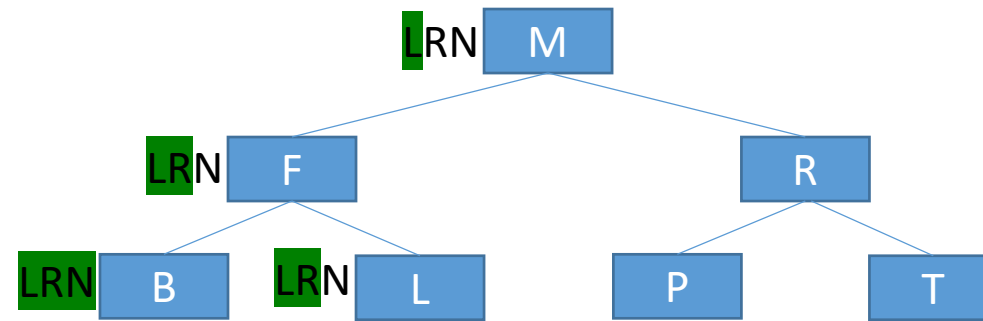
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B



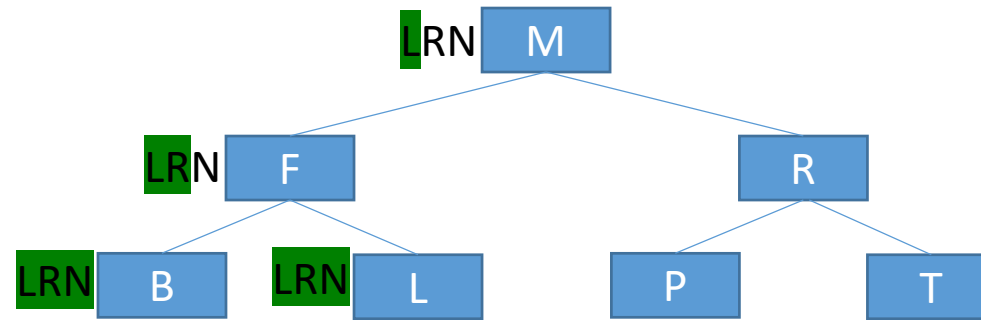
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B



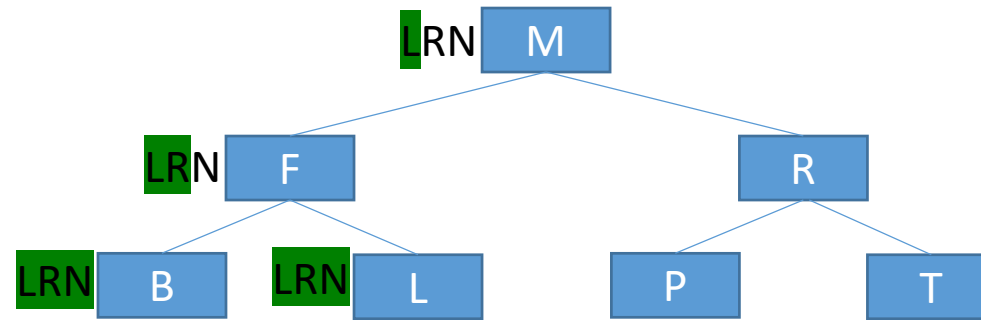
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B



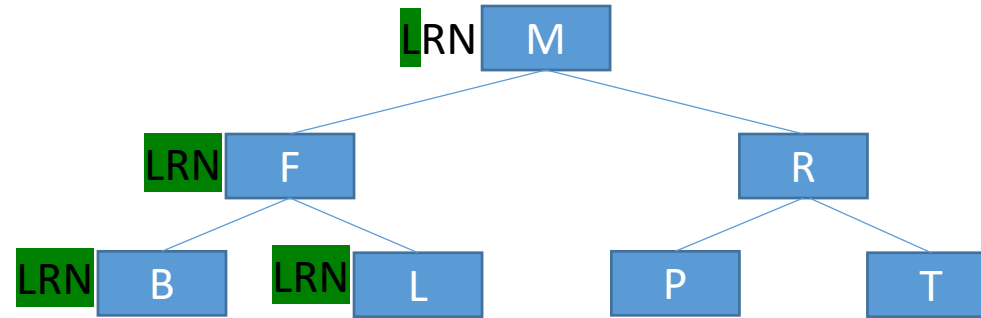
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B



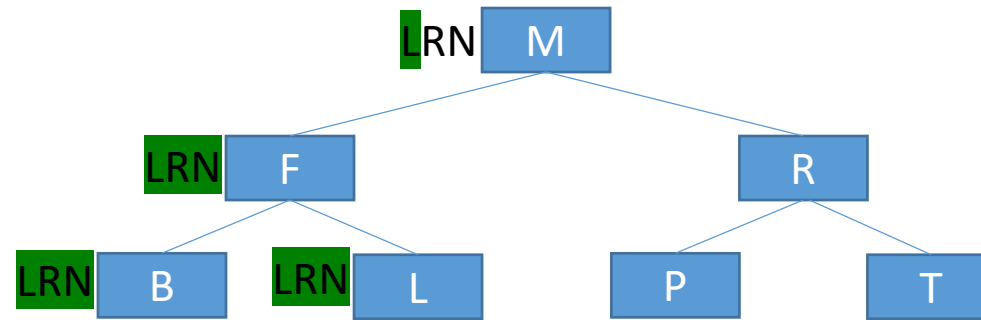
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B



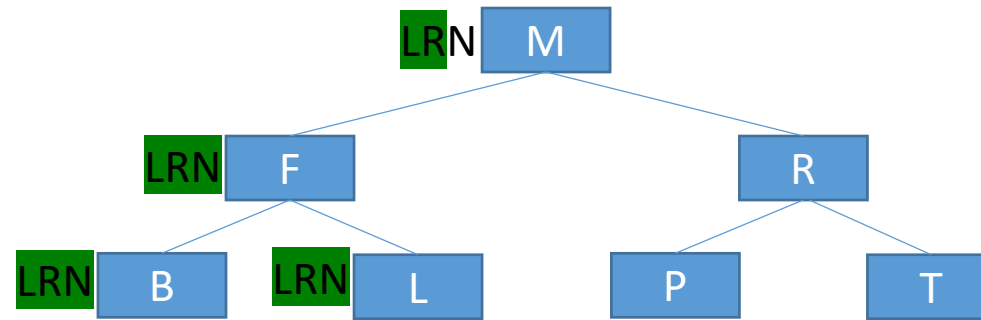
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L



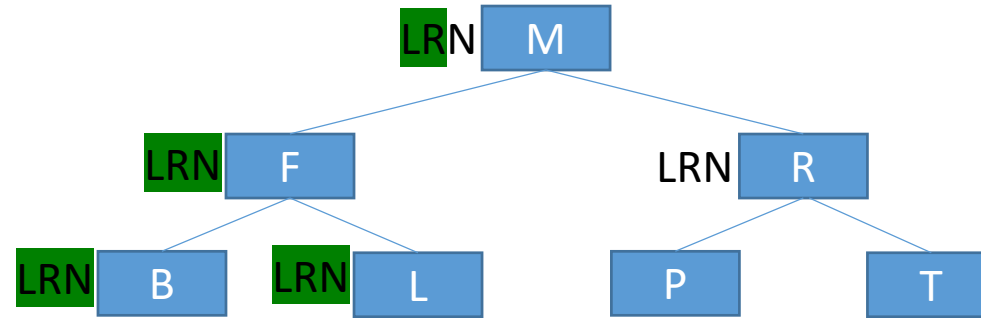
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L



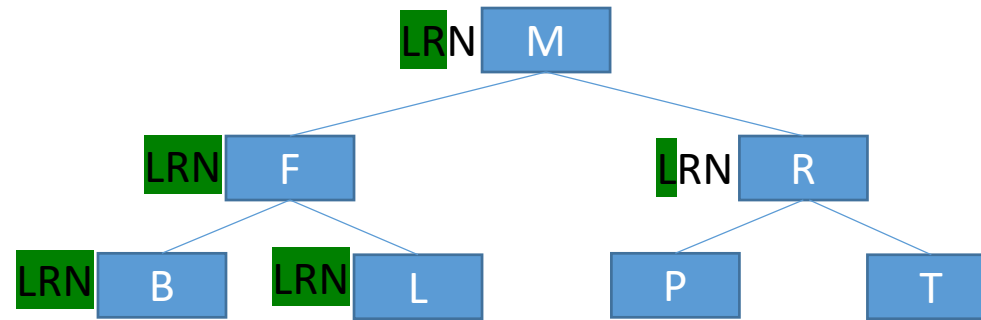
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



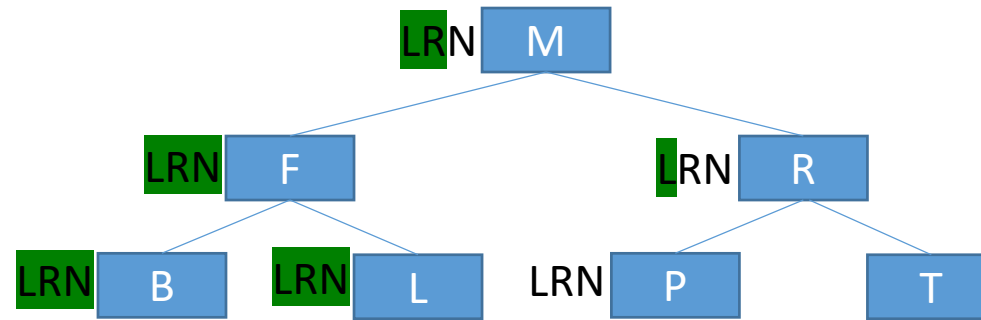
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



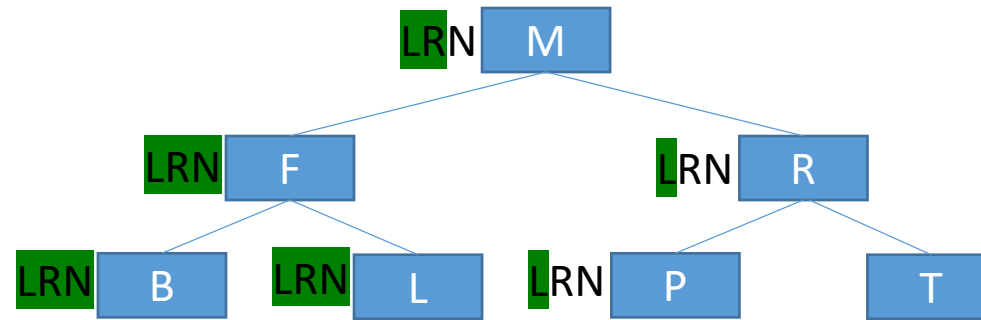
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



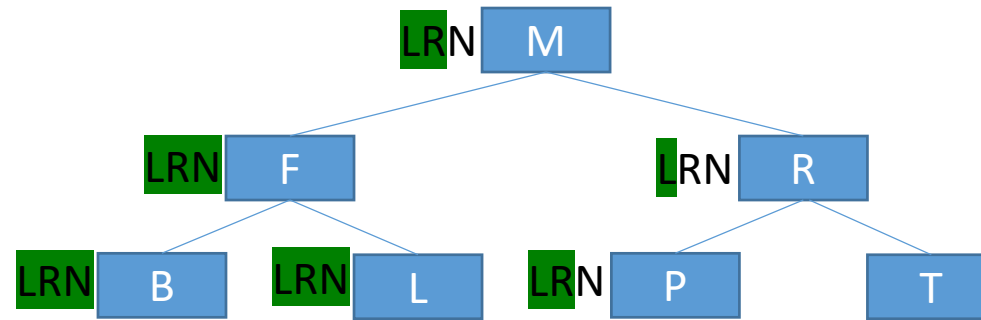
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



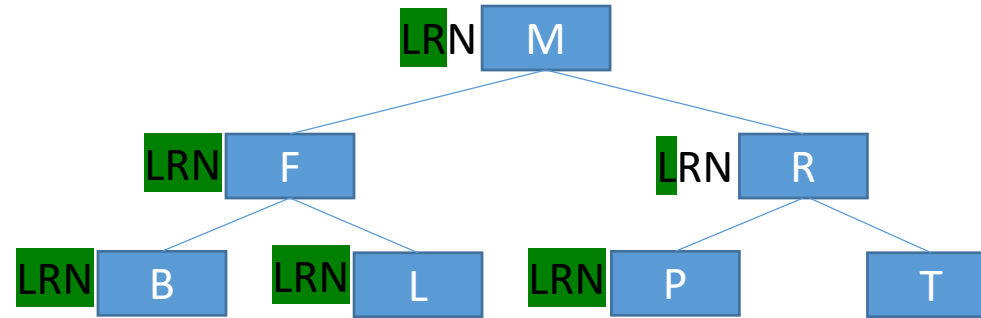
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



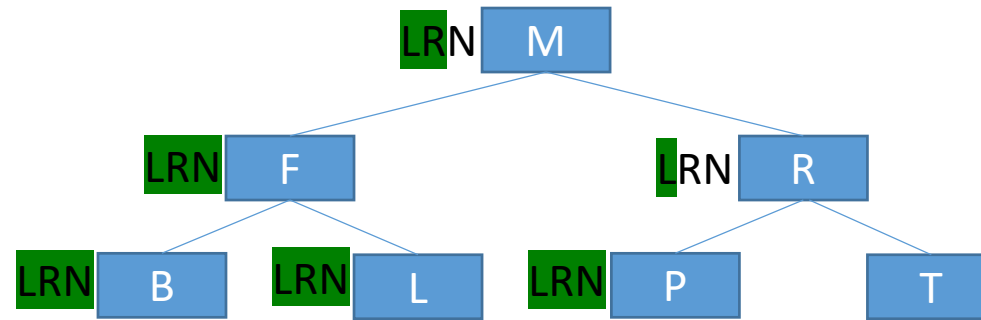
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



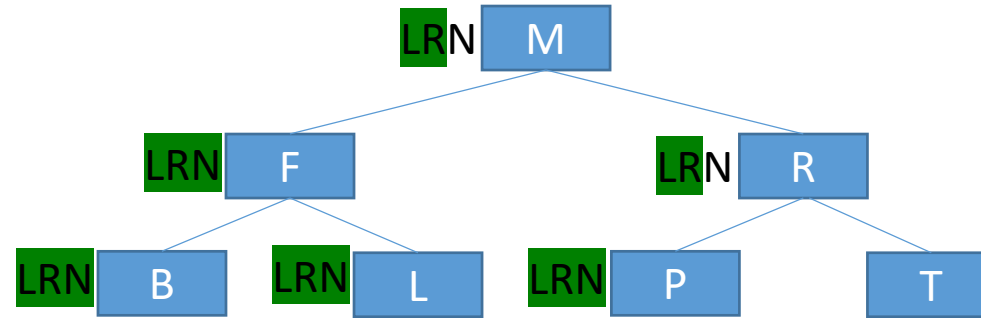
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



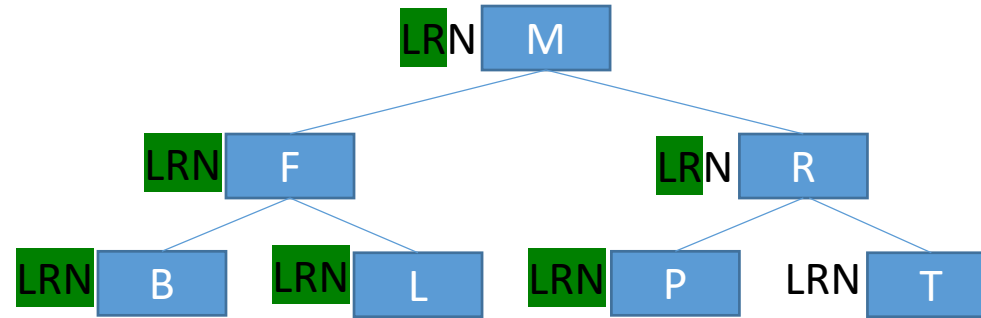
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F



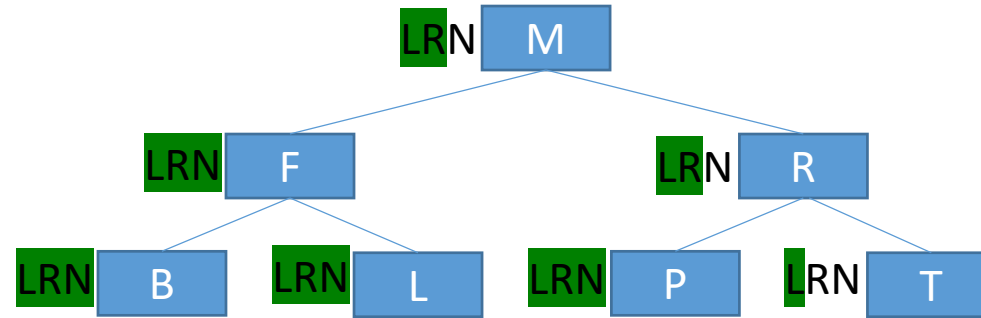
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P



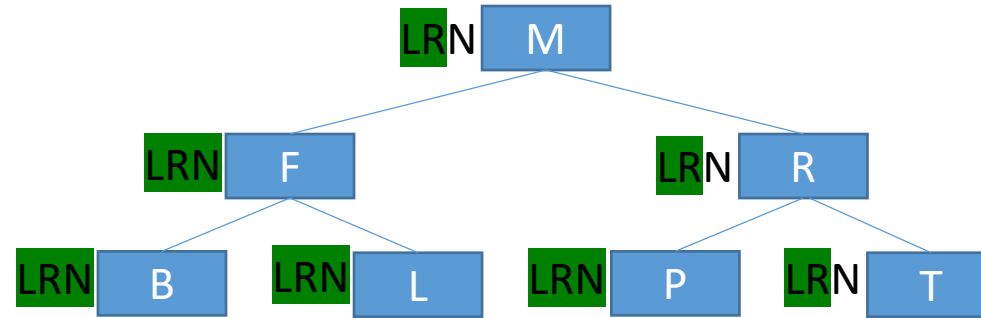
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P



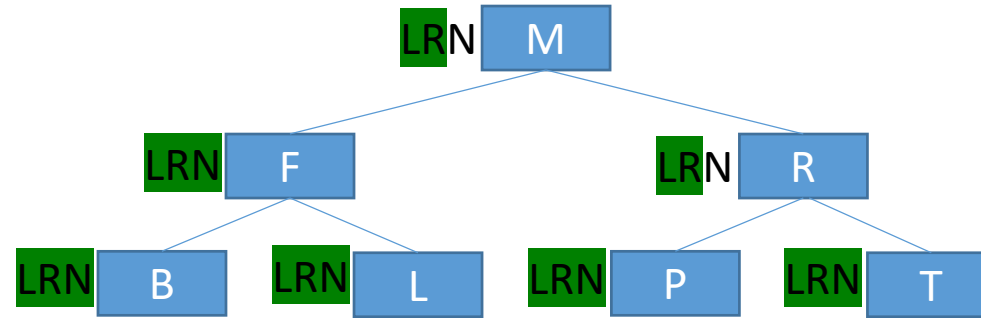
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P



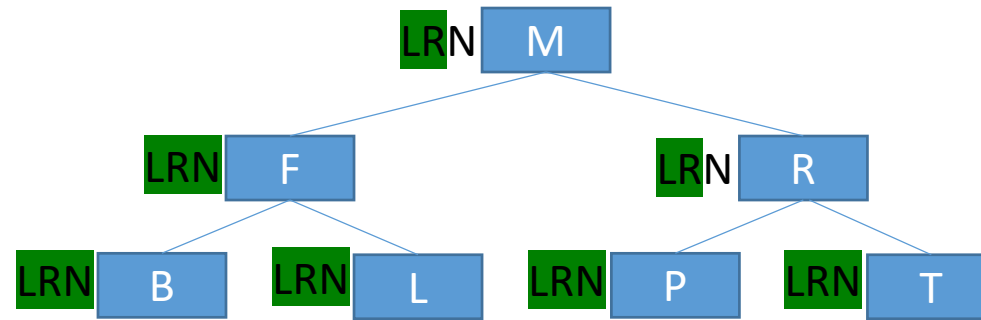
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P



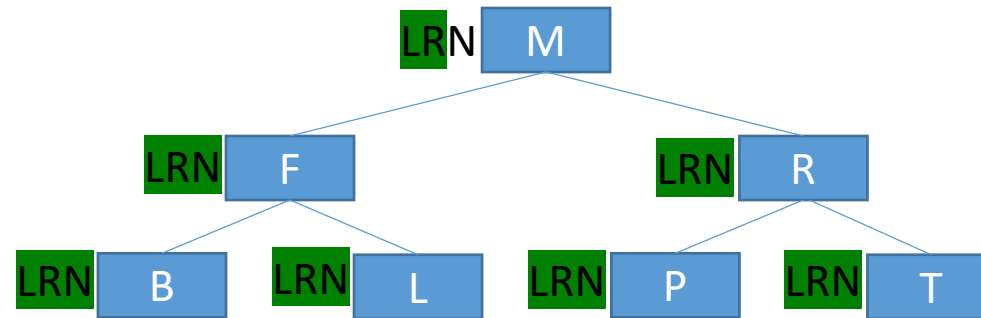
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P



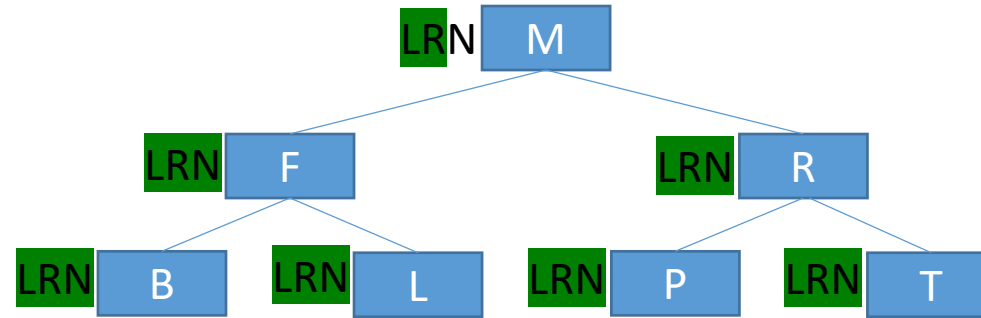
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P



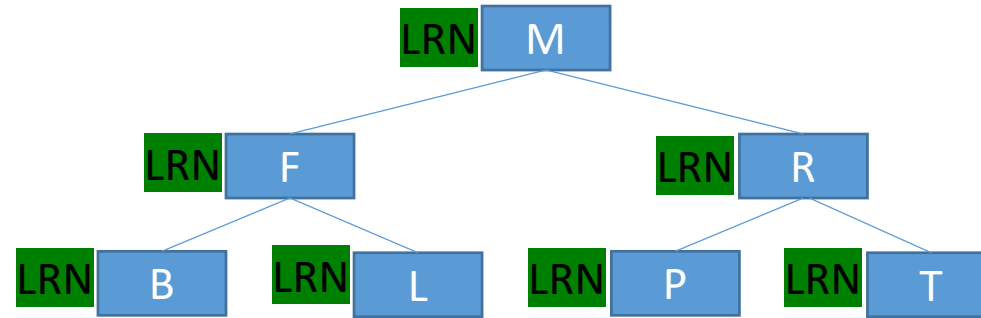
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T



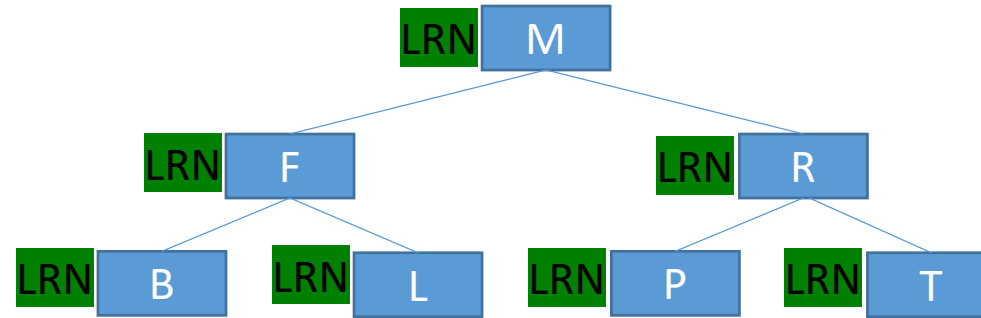
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T



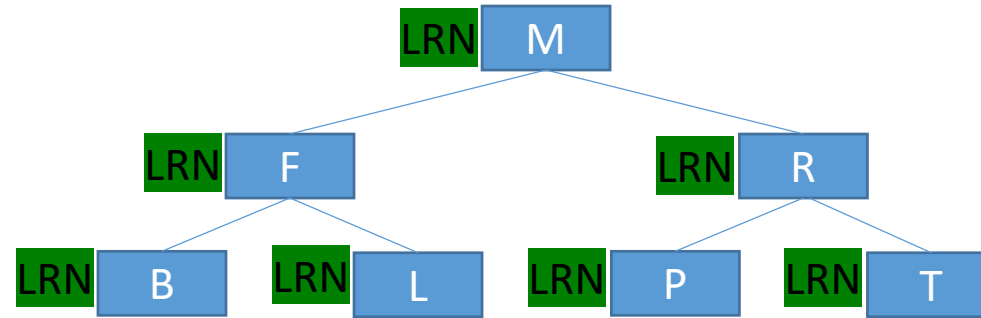
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R



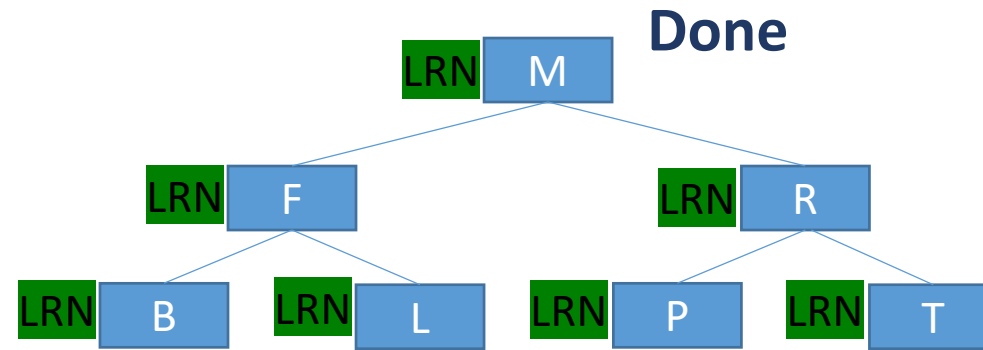
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R



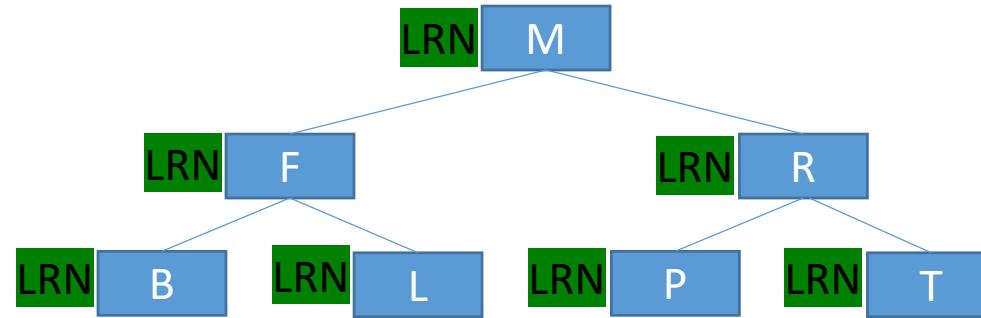
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- **Post-order:** B, L, F, P, T, R, M



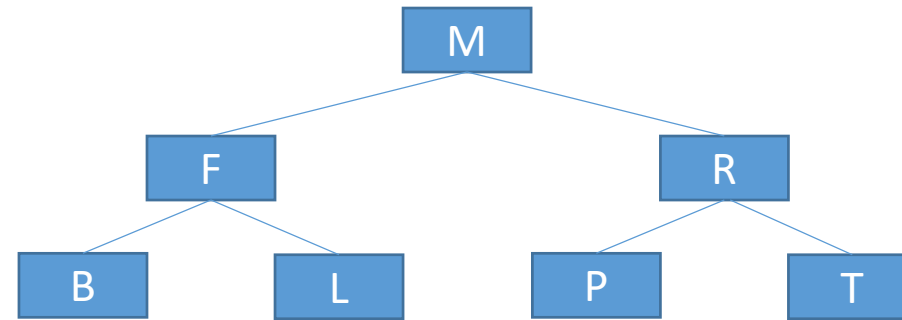
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R, M



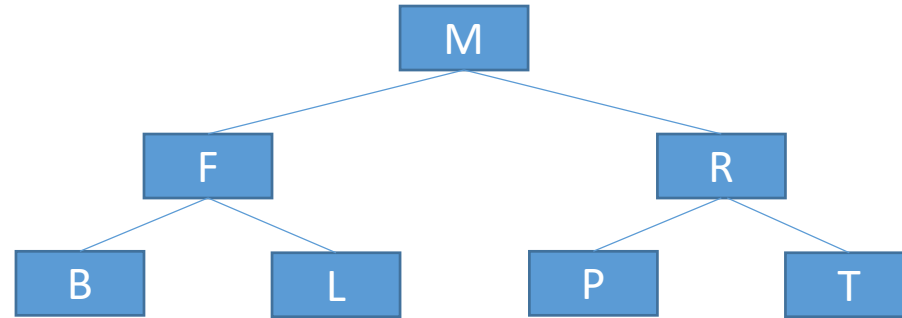
- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R, M



- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R, M



- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R, M



- Pre-order: M, F, B, L, R, P, T
- In-order: B, F, L, M, P, R, T
- Post-order: B, L, F, P, T, R, M