Graphs & LLRBs

Discussion 13

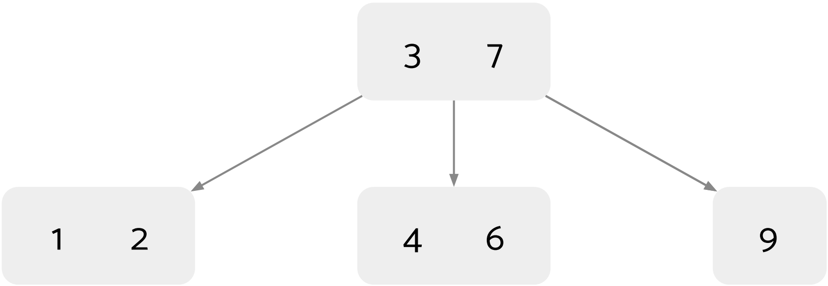
Announcements

- Week 13 Survey due Tuesday 04/19
- Homework 8 due Tuesday 04/19
- Lab 14 due Friday 04/22
- Project 3 Checkpoint due Friday 04/22
- Project 3 due Friday 04/29

Review

B-Trees

B-Trees (also referred to as 2-3 Trees) are trees that serve a similar function to binary trees while ensuring a bushy structure.

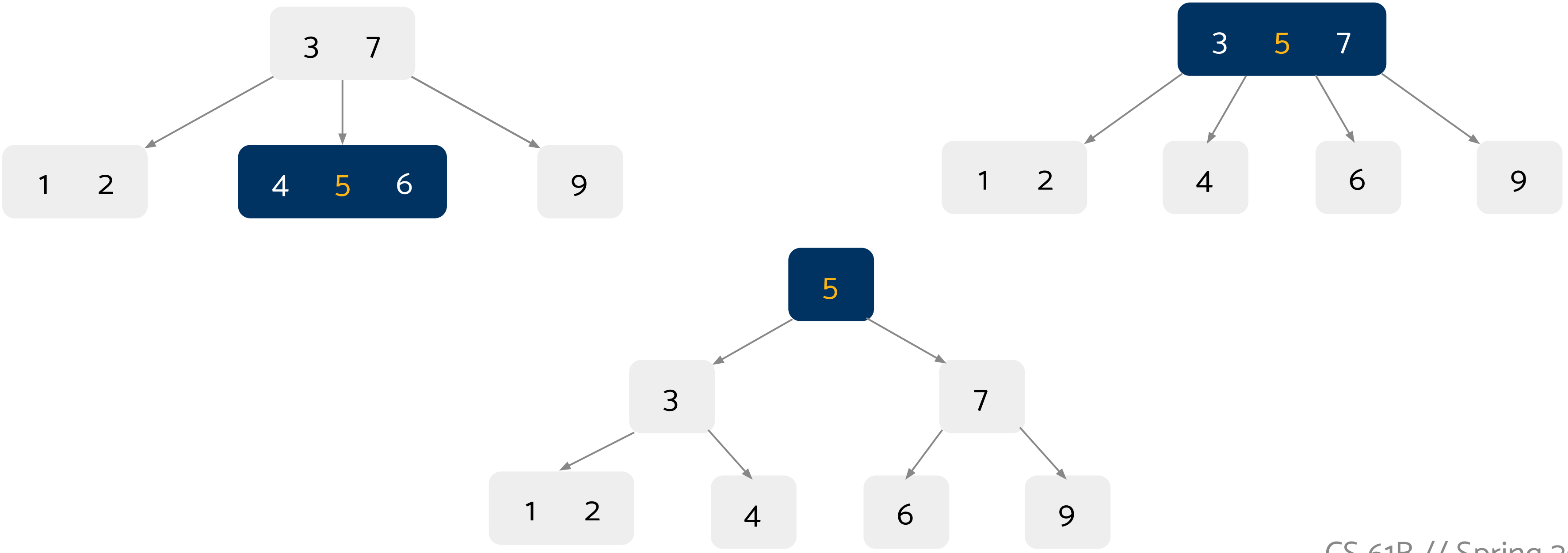


Each node can have up to 2 items and 3 children (there are variations where these values are higher known as 2-3-4 trees).

All leaves are the same distance from the root, which makes getting take Θ(log N) time.

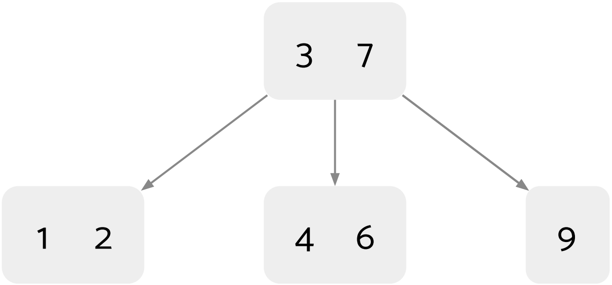
B-Trees

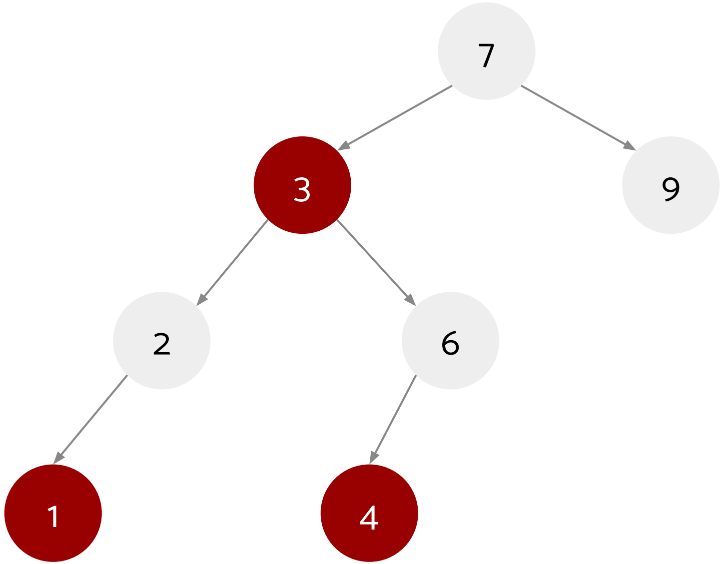
When adding to a B-Tree, you first start by adding to a leaf node, and then pushing the excess items up the tree until it follows the rules from the last slide.



Left Leaning Red Black Trees

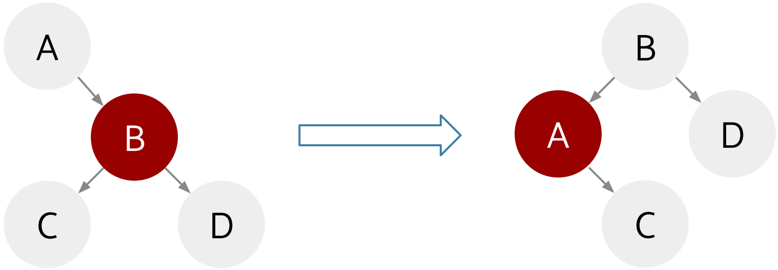
LLRBs are a representation of B-trees that we use because it is easier to work with in code. In an LLRB, each multi-node in a 2-3 tree is represented using a red connection on the left side.



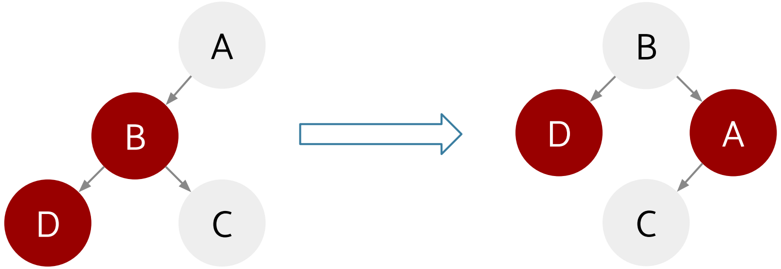


LLRB Balancing Operations

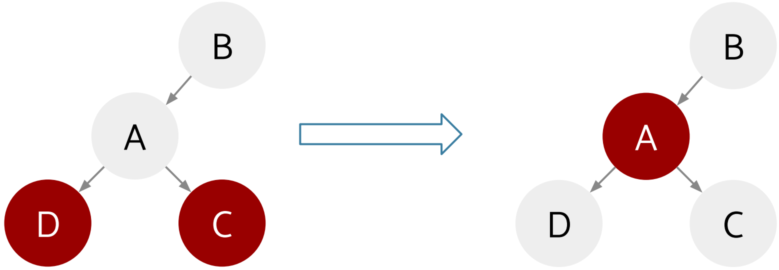
`rotateLeft(A);`



`rotateRight(A);`



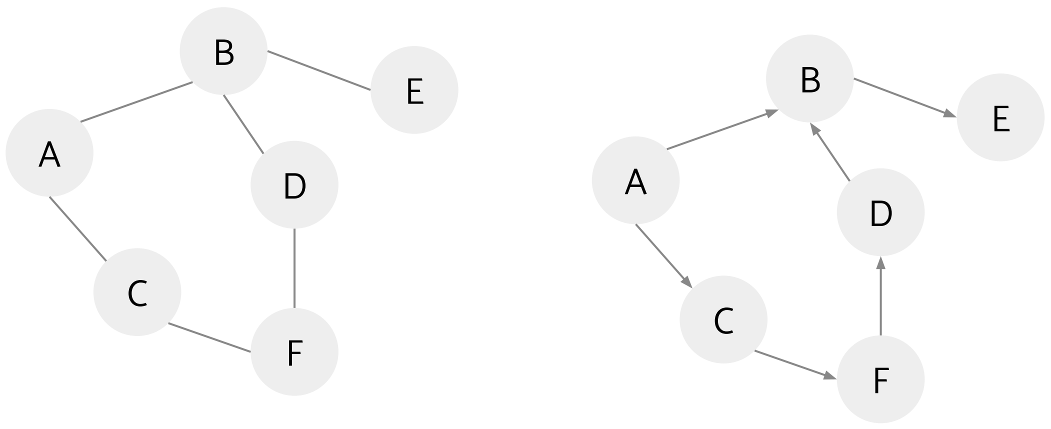
`colorFlip(A);`



Graphs

Graphs are structures that contain nodes and edges.

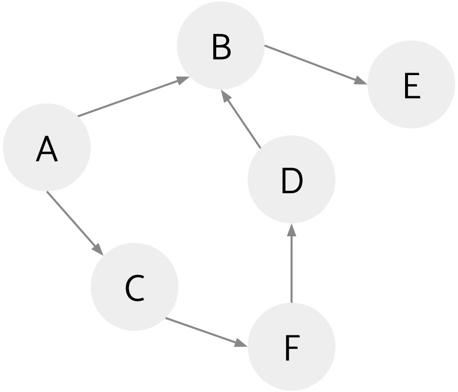
Graphs can be directed or undirected.



Graph Representations

Adjacency lists list out all the nodes connected to each node in our graph:

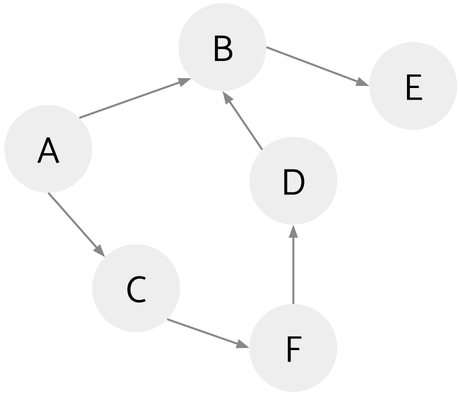
A	B , C
B	E
C	F
D	B
E	
F	D



Graph Representations

Adjacency matrices are true if there is a line going from node A to B and false otherwise.

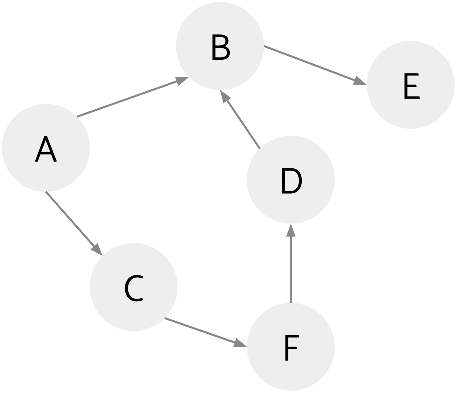
	A	B	C	D	E	F
A	0	1	1	0	0	0
B	0	0	0	0	1	0
C	0	0	0	0	0	1
D	0	1	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	1	0	0



Graph Searches

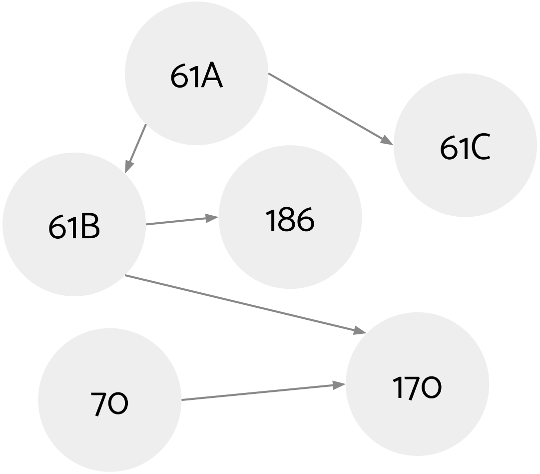
Breadth First Search goes in order of how far each node is from the starting node. Programmatically this is done using a queue.

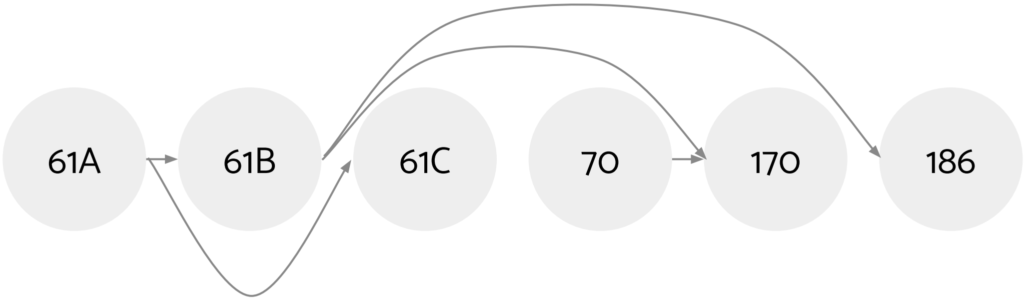
Depth First Search goes all the way down one path before exploring others. Programmatically this is done using a stack.



Topological Sorting

Topological sorting involves taking a graph with no cycles and laying it such that all of its edges point in the same direction. It can also be thought of as having every prerequisite physically to the left of each node.

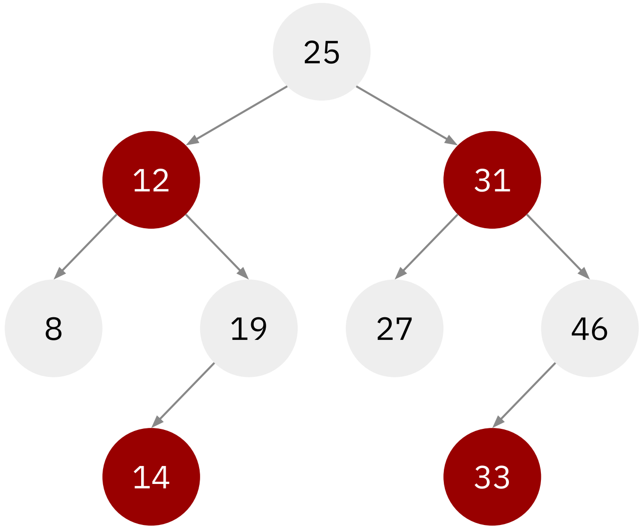




Worksheet

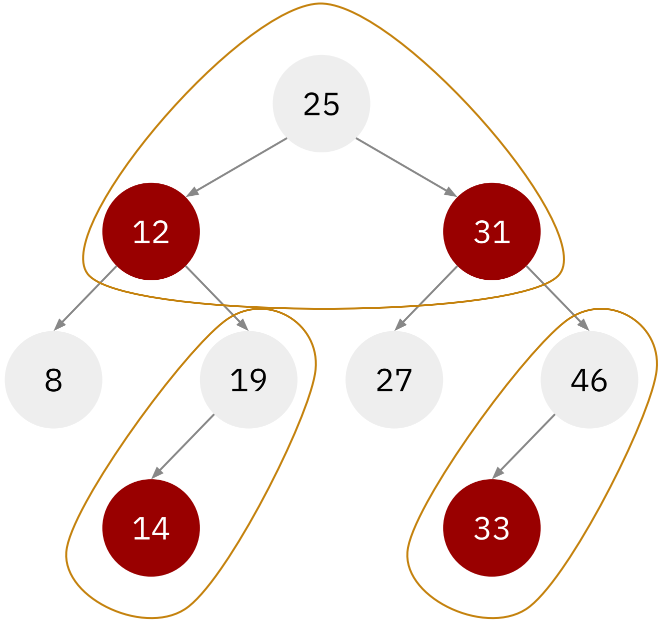
1A Balanced Search Trees

Convert the following red-black tree into a 2-4 Tree.



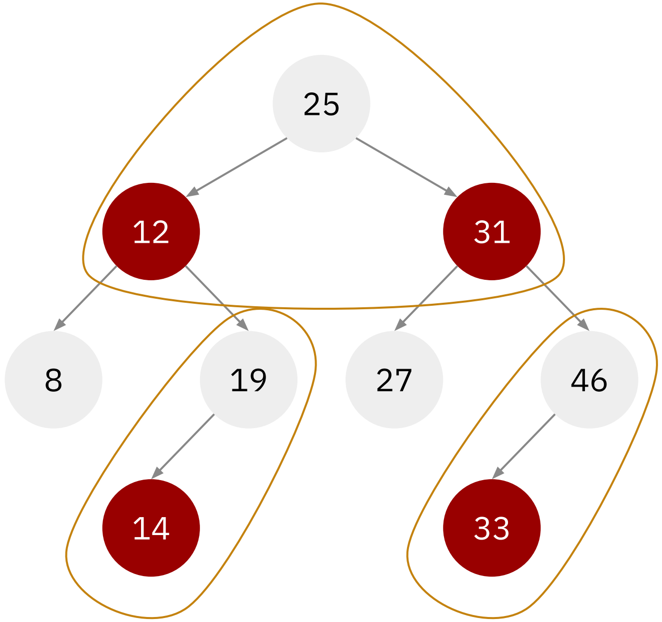
1A Balanced Search Trees

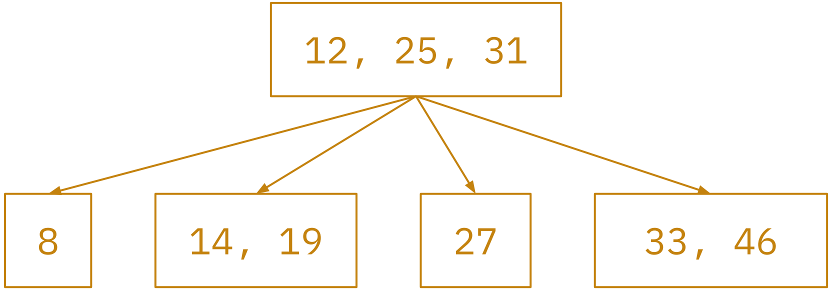
Convert the following red-black tree into a 2-4 Tree.



1A Balanced Search Trees

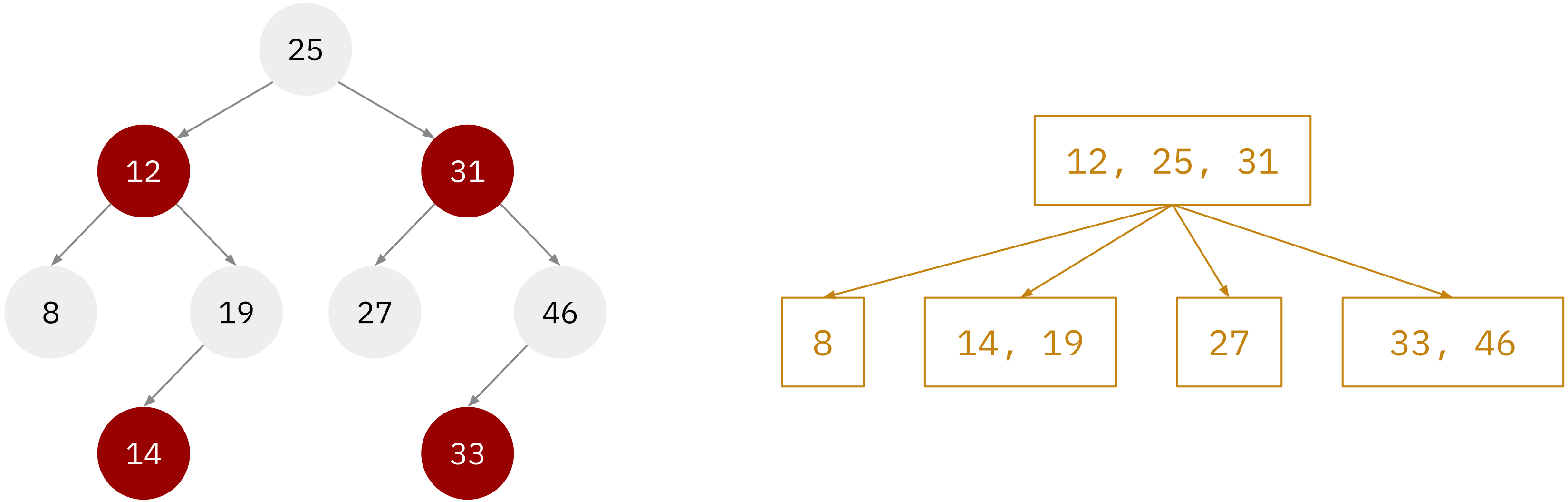
Convert the following red-black tree into a 2-4 Tree.





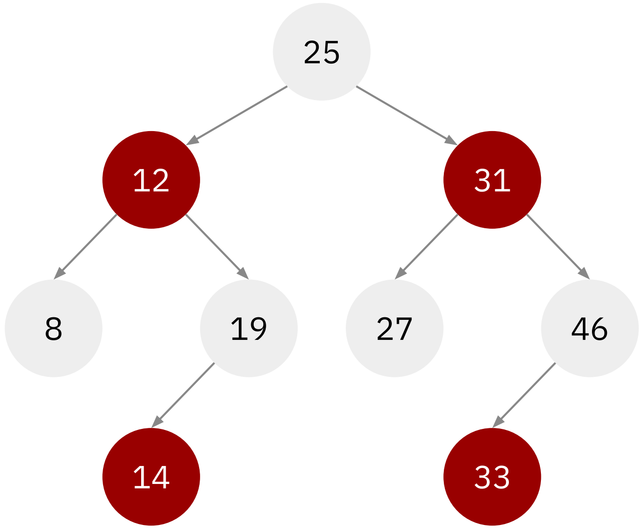
1B Balanced Search Trees

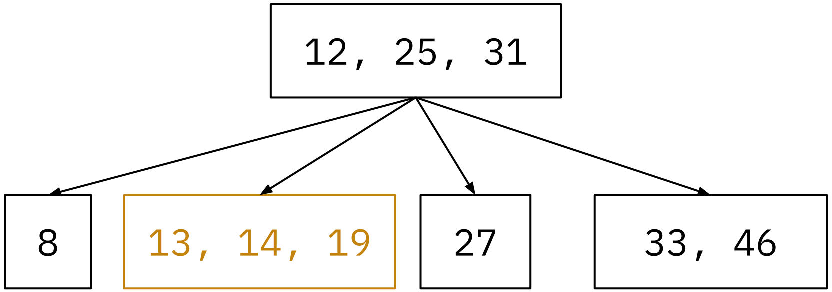
Insert 13 and 17 into your resulting 2-4 Tree.



1B Balanced Search Trees

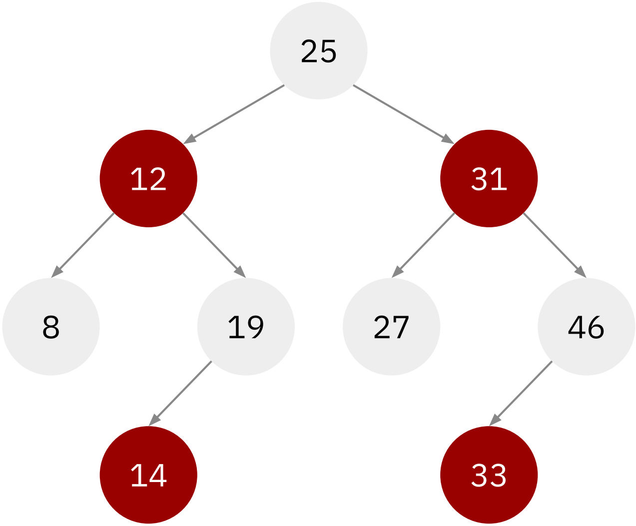
Insert 13 and 17 into your resulting 2-4 Tree.

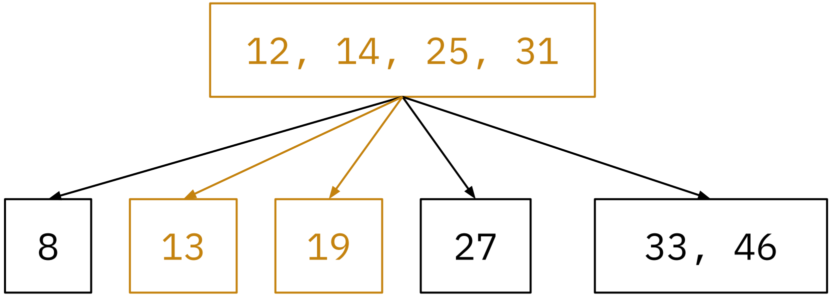




1B Balanced Search Trees

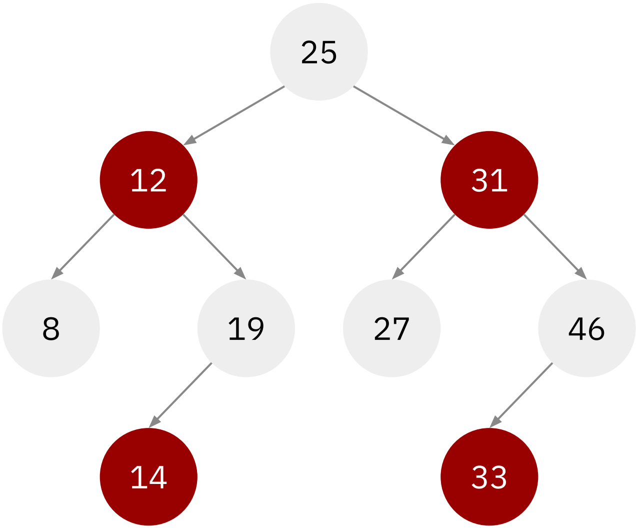
Insert 13 and 17 into your resulting 2-4 Tree.

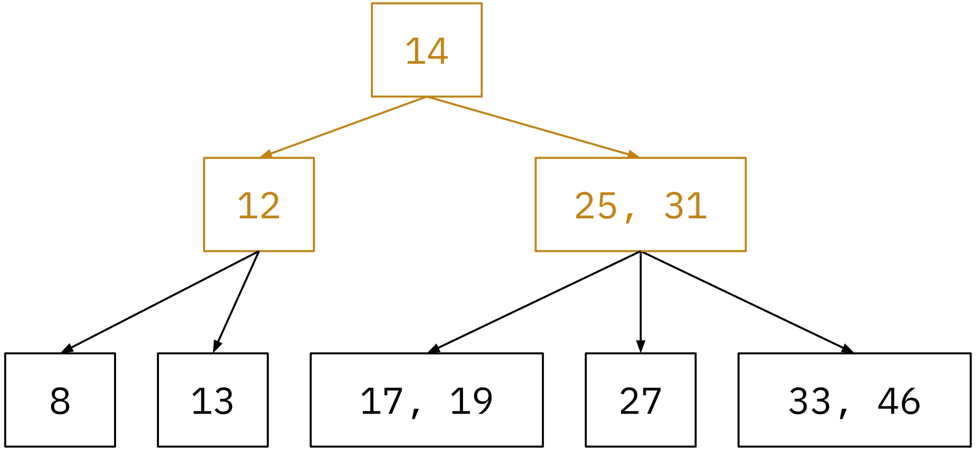




1B Balanced Search Trees

Insert 13 and 17 into your resulting 2-4 Tree.



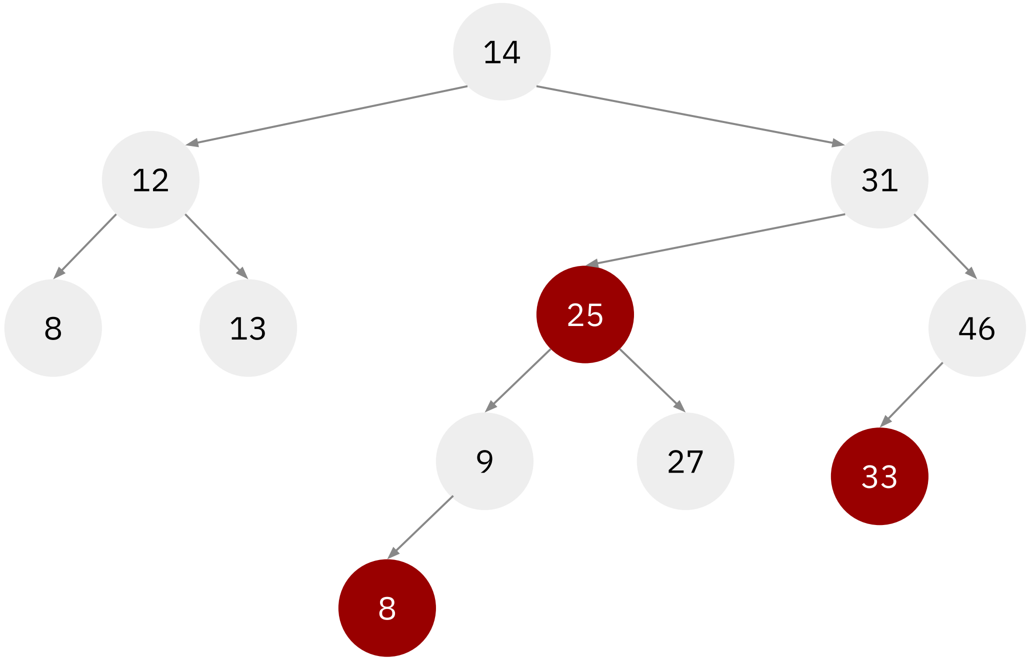


1C Balanced Search Trees

Convert your resulting 2-4 Tree into a valid LLRB.

1C Balanced Search Trees

Convert your resulting 2-4 Tree into a valid LLRB.



1D Balanced Search Trees

Given a 2-4 Tree with N keys, describe how you can obtain the keys in sorted order in O(N) time.

1D Balanced Search Trees

Given a 2-4 Tree with N keys, describe how you can obtain the keys in sorted order in O(N) time.

Generalize an in-order traversal - traverse the left, emit the first key of the node, traverse the second child of the nose, emit the second key of the node, etc.

1E Balanced Search Trees

If a 2-4 Tree has depth H, what is the maximum number of comparisons done in the corresponding LLRB to find a certain key.

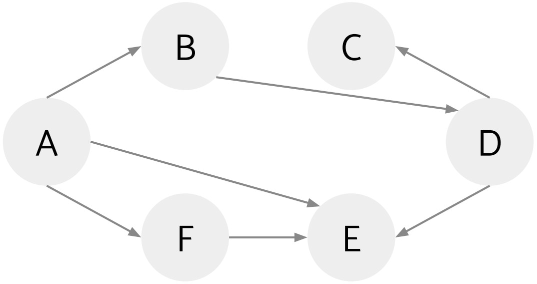
1E Balanced Search Trees

If a 2-4 Tree has depth H, what is the maximum number of comparisons done in the corresponding LLRB to find a certain key.

2H comparisons.

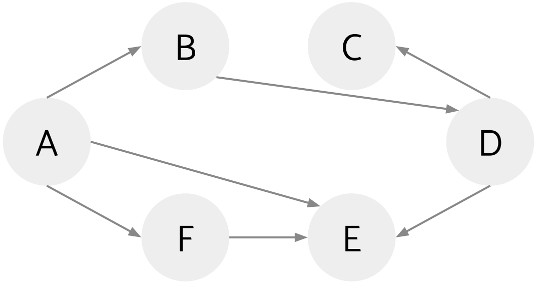
2 Graph Representation

Represent the graph with an adjacency list and matrix representation.



2 Graph Representation

Represent the graph with an adjacency list and matrix representation.

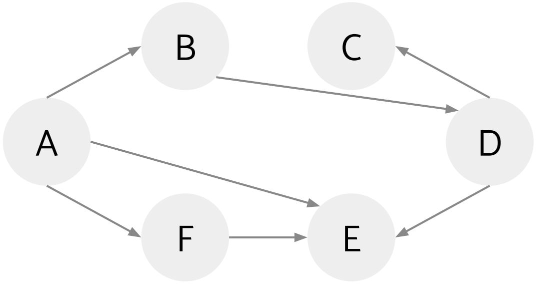



```
Adjacency List
A -> [B, E, F]
B -> [D]
C -> []
D -> [C, E]
E -> []
F -> [E]

TO
  A B C D E F
A 0 1 0 0 1 1
B 0 0 0 1 0 0
C 0 0 0 0 0 0
D 0 0 1 0 1 0
E 0 0 0 0 0 0
F 0 0 0 0 1 0
```

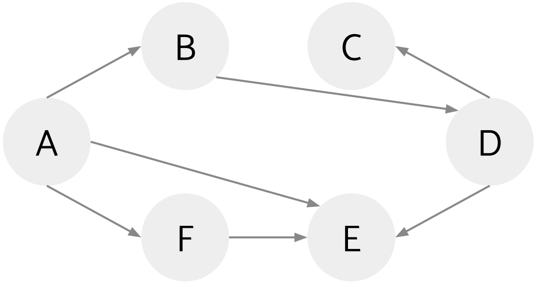
3 Searches and Traversals

Run DFS preorder, DFS postorder, and BFS on the graph starting from node A.



3 Searches and Traversals

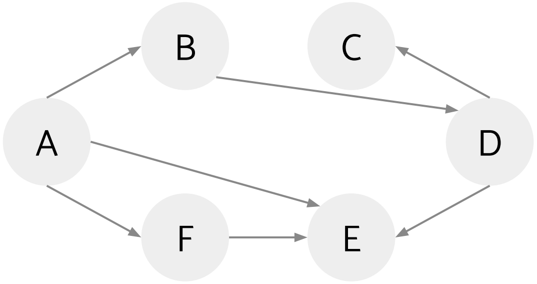
Run DFS preorder, DFS postorder, and BFS on the graph starting from node A.




```
DFS Preorder:  A, B, D, C, E, F
```

3 Searches and Traversals

Run DFS preorder, DFS postorder, and BFS on the graph starting from node A.

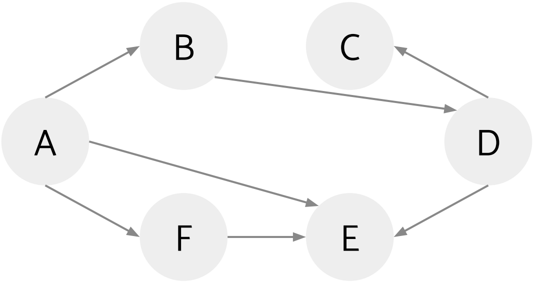


`DFS Postorder: C, E, D, B, F, A`

`BFS: A, B, E, F, D, C`

3 Searches and Traversals

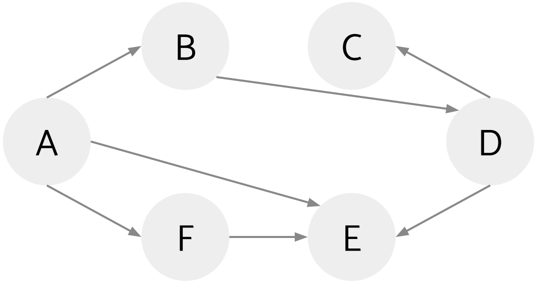
Run DFS preorder, DFS postorder, and BFS on the graph starting from node A.



`BFS: A, B, E, F, D, C`

4 Topological Sorting

Give a valid topological order of the graph. Is it unique?



4 Topological Sorting

Give a valid topological order of the graph. Is it unique?