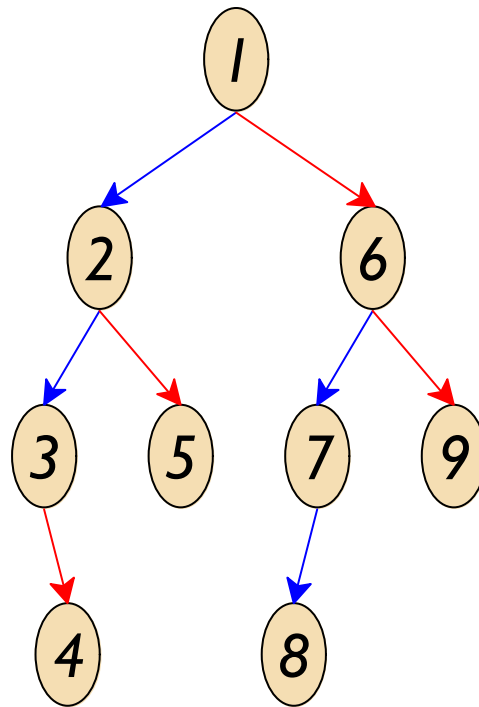
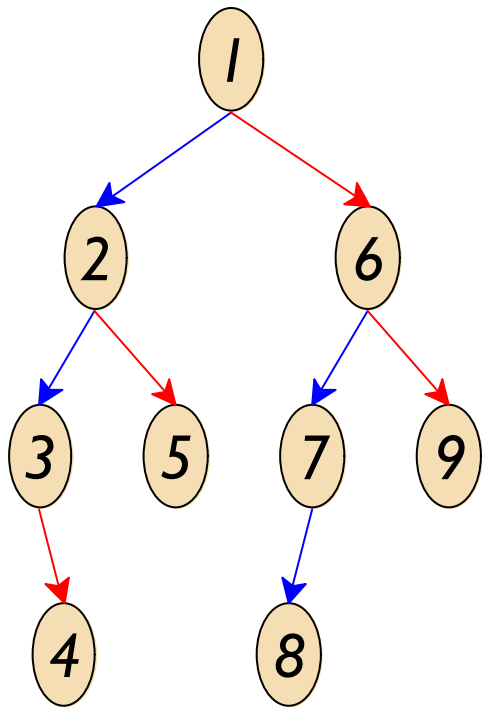


Depth-First Traversal



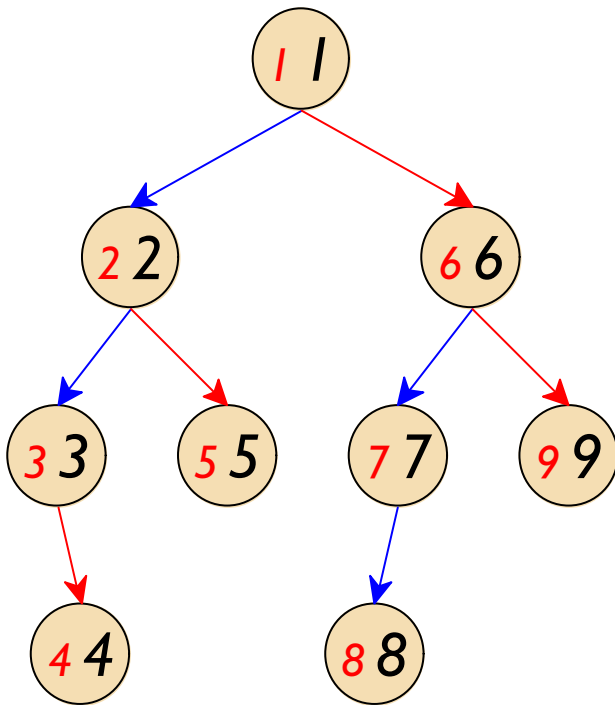
Depth-First Traversal



```
(define (depth-first t)
  (cond
    [(empty? t) ....]
    [else
     .... (node-val t) ....
     (depth-first (node-left t))
     (depth-first (node-right t))]))
```

Depth-First Traversal — Preorder

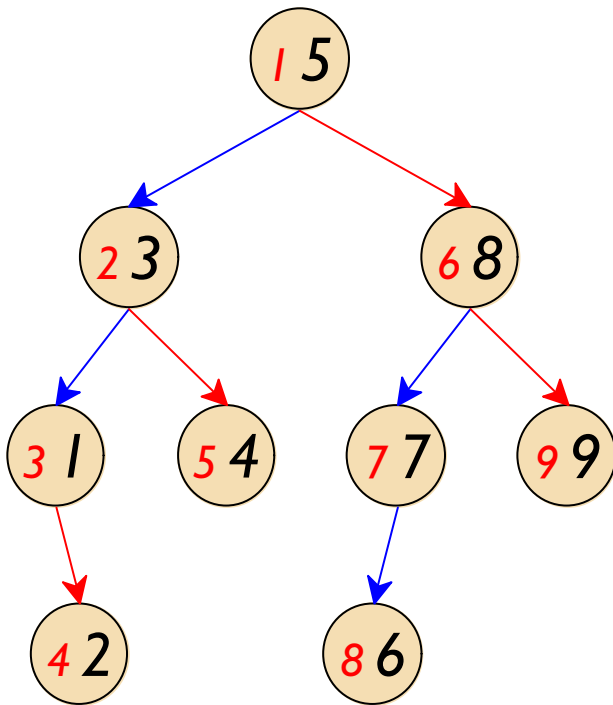
Preorder visits each node before its children



```
(define (preorder t visit)
  (cond
    [(empty? t) (void)]
    [else
     (visit (node-val t))
     (preorder (node-left t)
               visit)
     (preorder (node-right t)
               visit)]))
```

Depth-First Traversal — Inorder

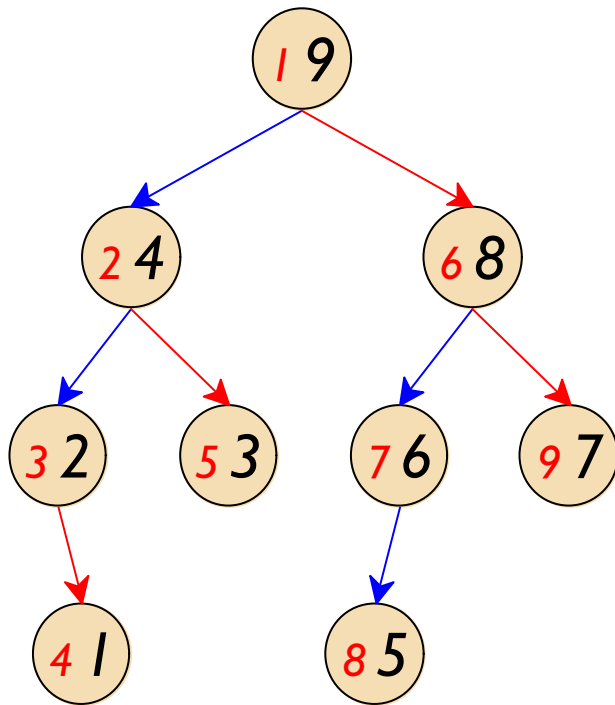
Inorder visits left children first, right children last



```
(define (inorder t visit)
  (cond
    [(empty? t) (void)]
    [else
     (inorder (node-left t)
              visit)
     (visit (node-val t))
     (inorder (node-right t)
              visit)]))
```

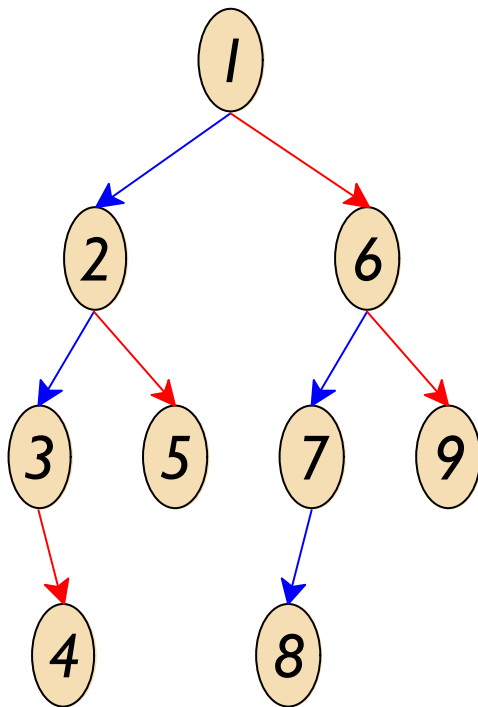
Depth-First Traversal — Postorder

Postorder visits each node after its children



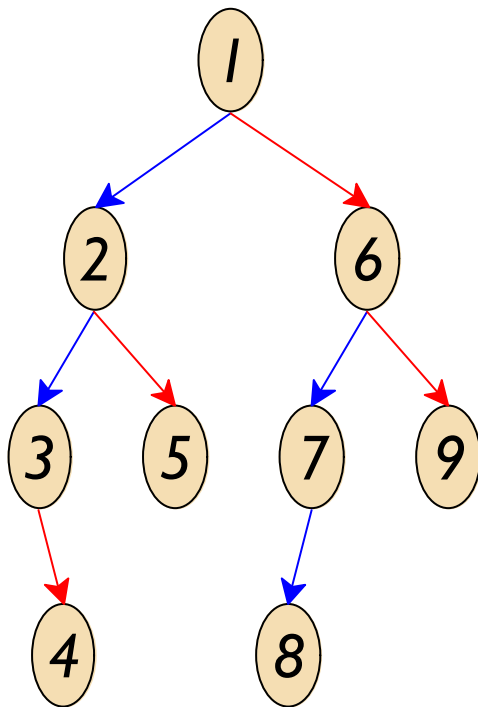
```
(define (postorder t visit)
  (cond
    [(empty? t) (void)]
    [else
     (postorder (node-left t)
                visit)
     (postorder (node-right t)
                visit)
     (visit (node-val t))]))
```

Depth-First Traversal and Loops



```
void depth_first(tree t) {  
    while (t != NULL) {  
        visit(t->val);  
        t = t->???  
    }  
}
```

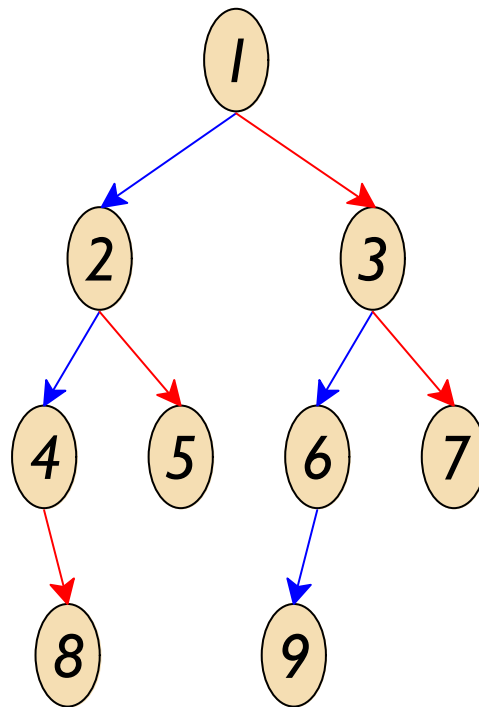
Depth-First Traversal and Loops



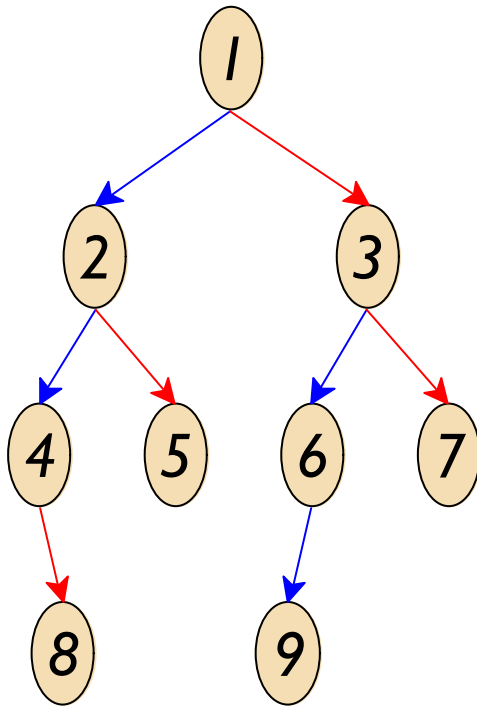
```
void depth_first(tree t) {  
    while (t != NULL) {  
        visit(t->val);  
        t = t->??;  
    }  
}
```

A depth-first traversal doesn't fit directly into a loop

Breadth-First Traversal

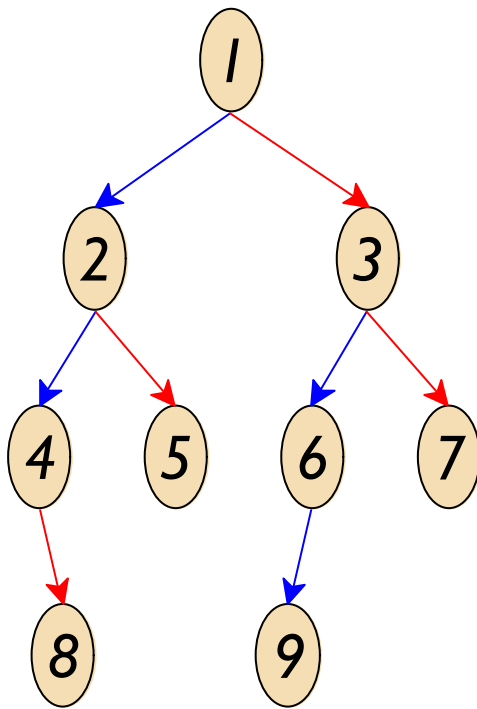


Breadth-First Traversal



```
(define (breadth-first t)
  (cond
    [(empty? t) ....]
    [else
     .... (node-val t) ....
     ; ? explores left tree ?
     (breadth-first (node-left t))
     (breadth-first (node-right t))]))
```

Breadth-First Traversal



```
(define (breadth-first t)
  (cond
    [(empty? t) ....]
    [else
     .... (node-val t) ....
     ; ? explores left tree ?
     (breadth-first (node-left t))
     (breadth-first (node-right t))]))
```

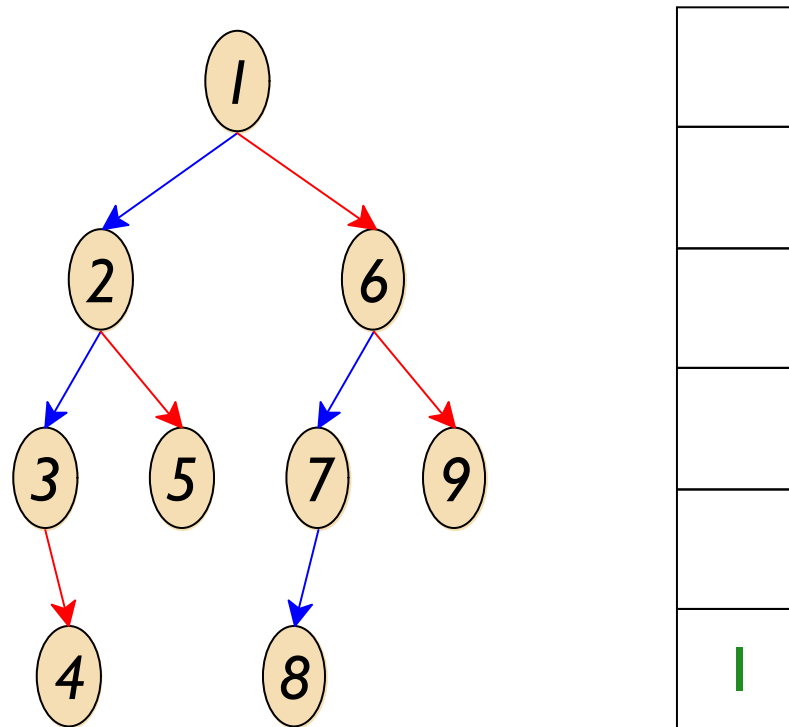
A depth-first traversal doesn't fit directly into recursion

Traversals as Loops

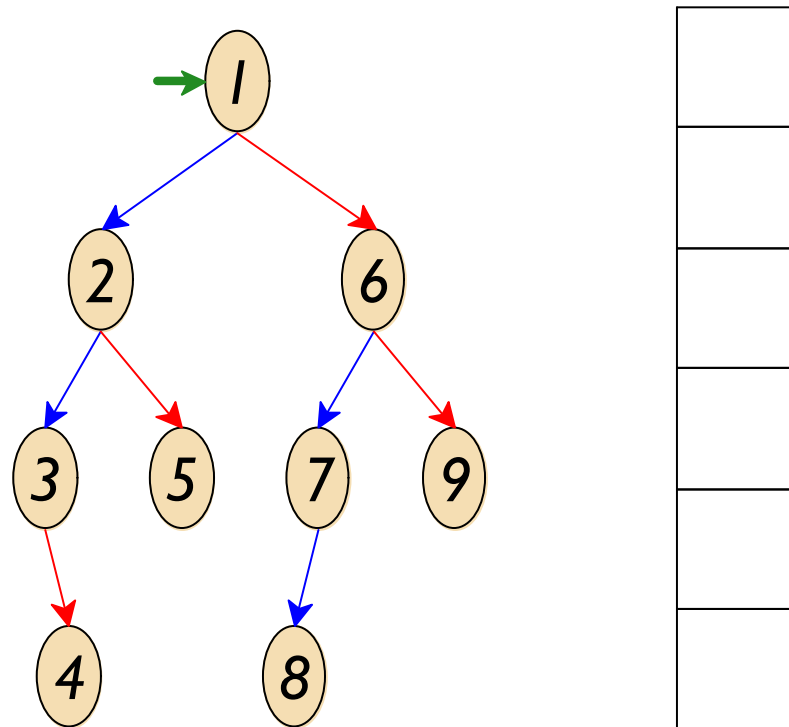
To turn a tree traversal in a loop, use a container to keep track of nodes yet to be visited

- **stack** container \Rightarrow **depth-first** traversal
- **queue** container \Rightarrow **breadth-first** traversal

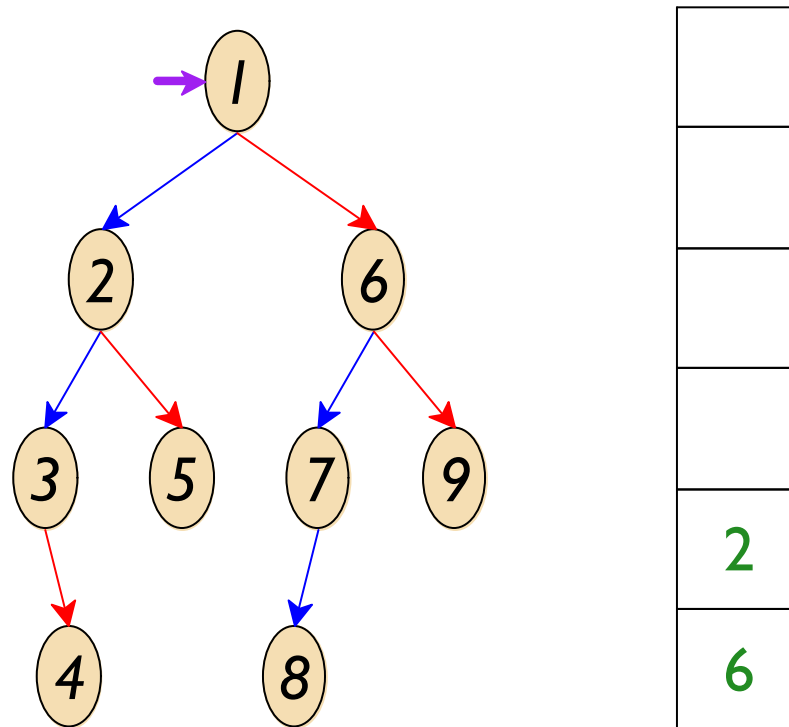
Depth-First Traversal (Preorder) via Stack



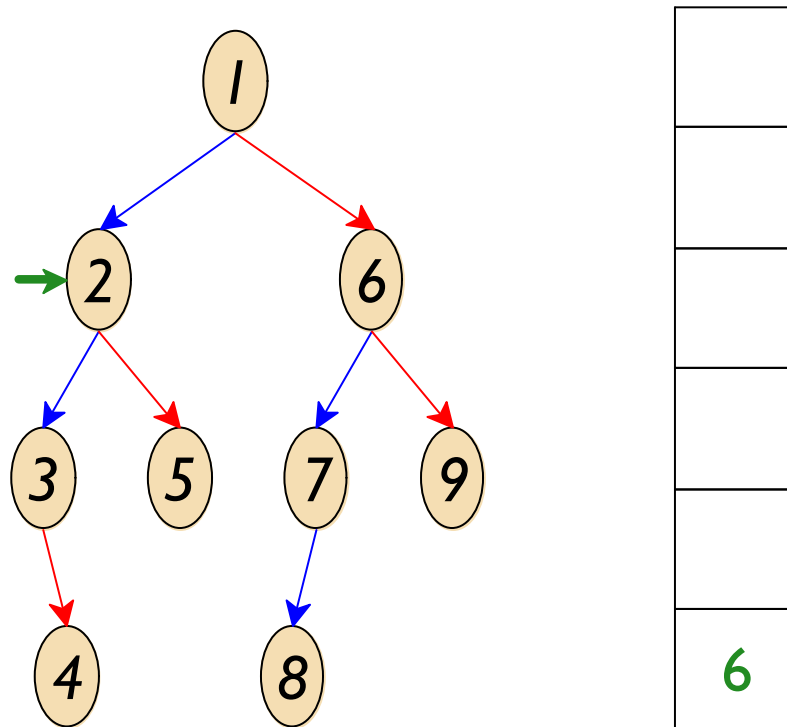
Depth-First Traversal (Preorder) via Stack



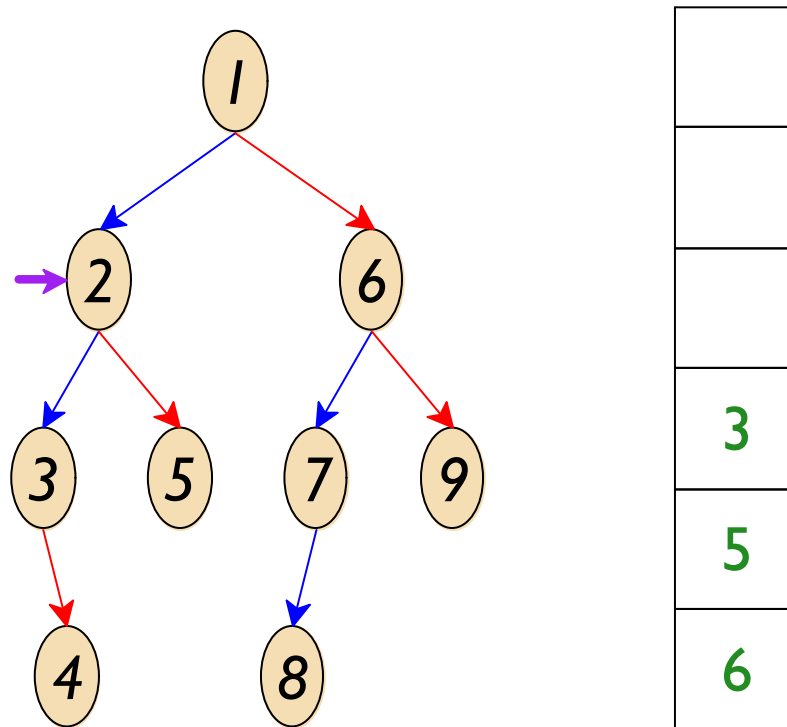
Depth-First Traversal (Preorder) via Stack



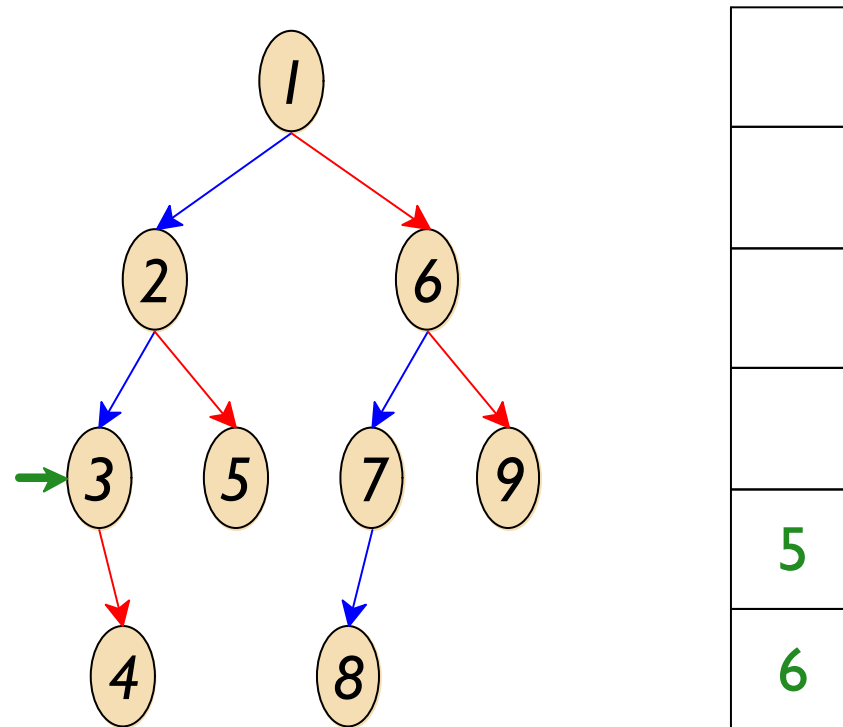
Depth-First Traversal (Preorder) via Stack



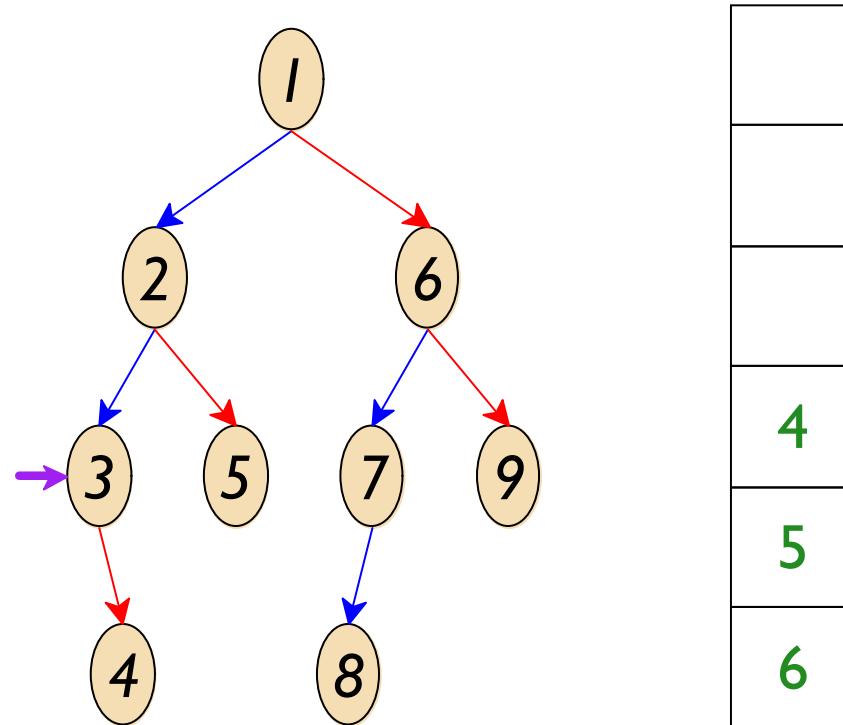
Depth-First Traversal (Preorder) via Stack



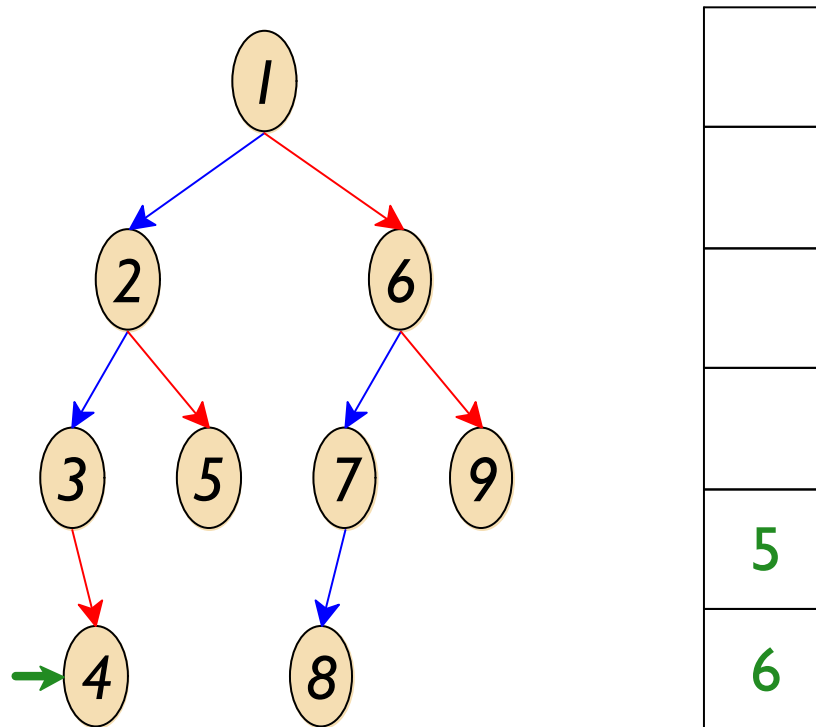
Depth-First Traversal (Preorder) via Stack



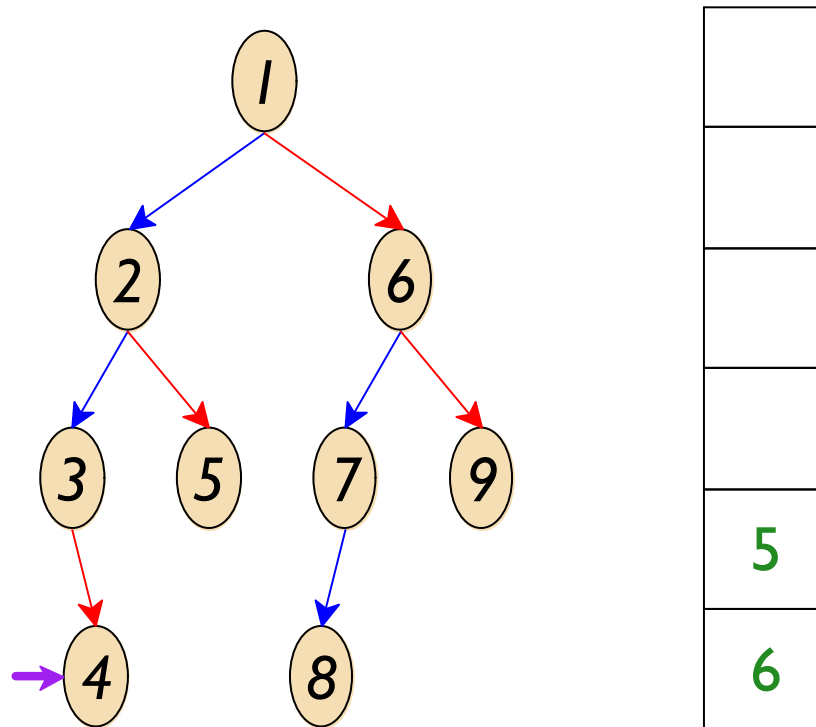
Depth-First Traversal (Preorder) via Stack



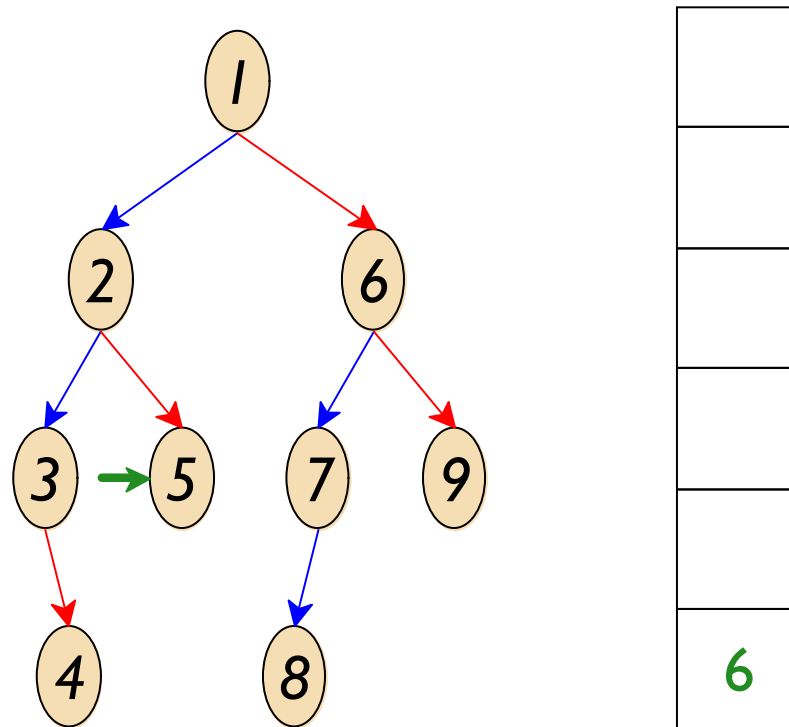
Depth-First Traversal (Preorder) via Stack



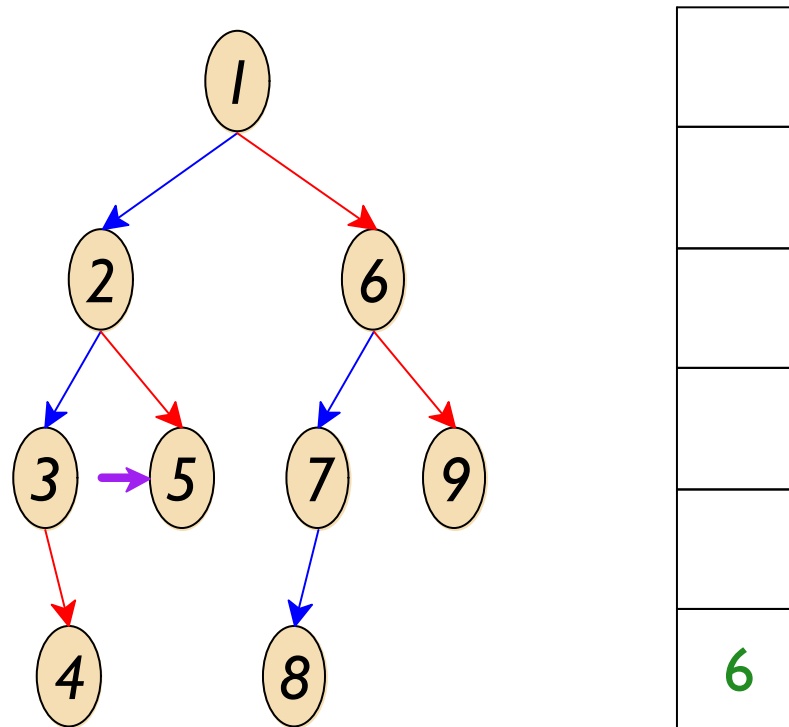
Depth-First Traversal (Preorder) via Stack



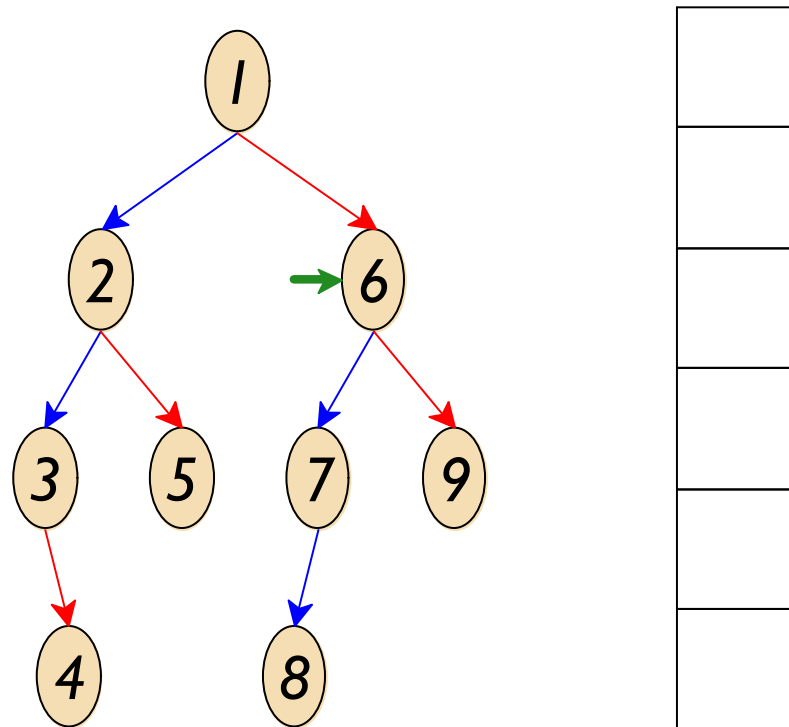
Depth-First Traversal (Preorder) via Stack



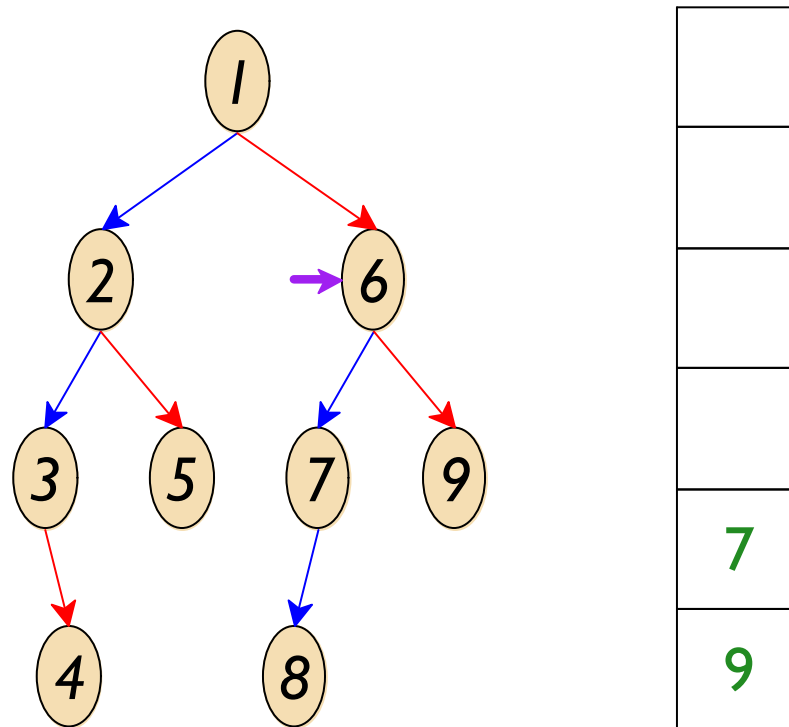
Depth-First Traversal (Preorder) via Stack



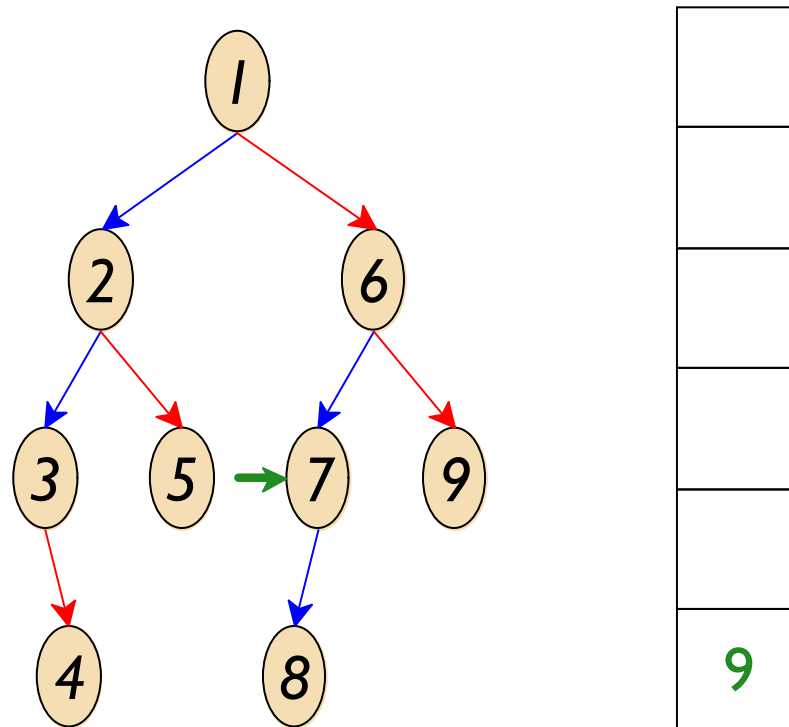
Depth-First Traversal (Preorder) via Stack



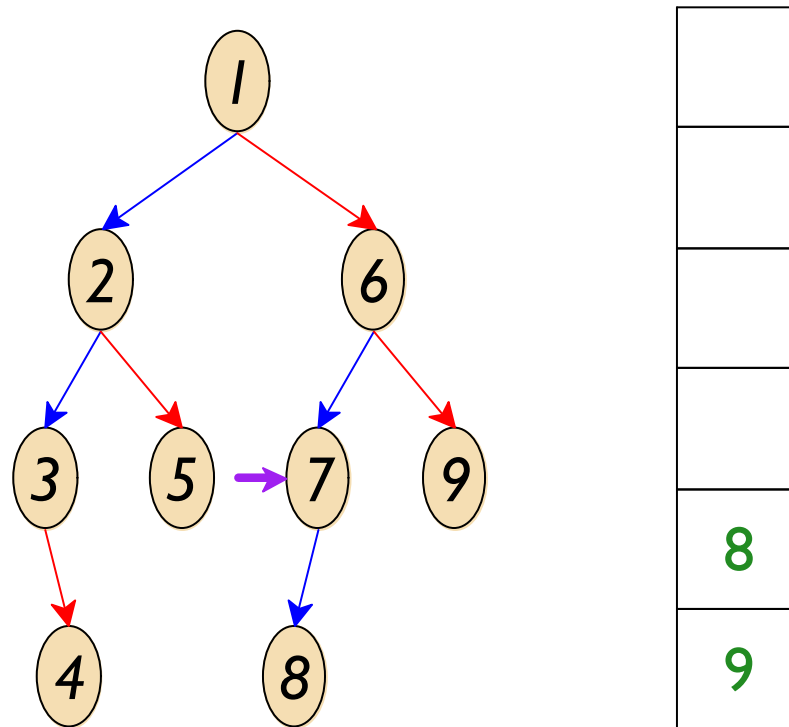
Depth-First Traversal (Preorder) via Stack



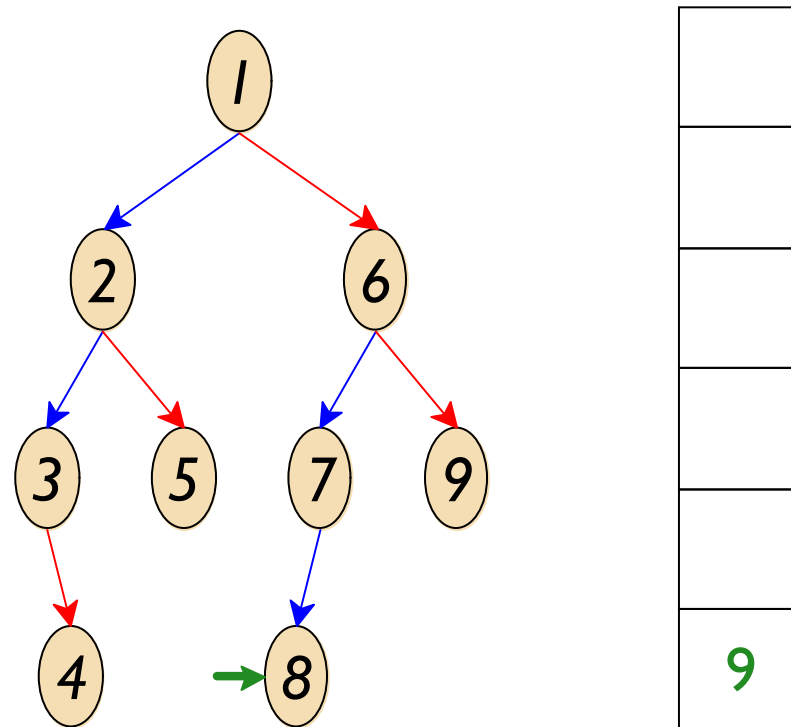
Depth-First Traversal (Preorder) via Stack



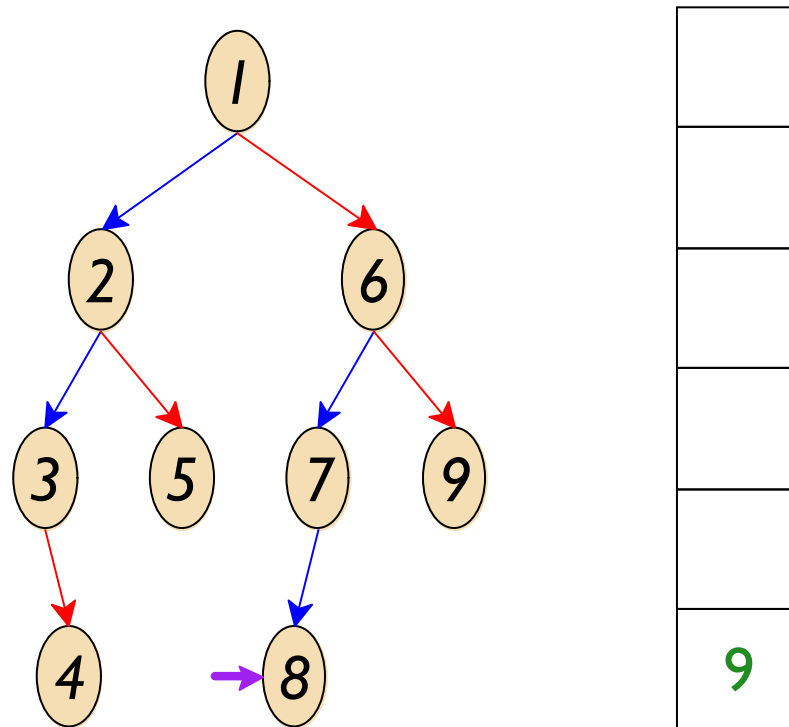
Depth-First Traversal (Preorder) via Stack



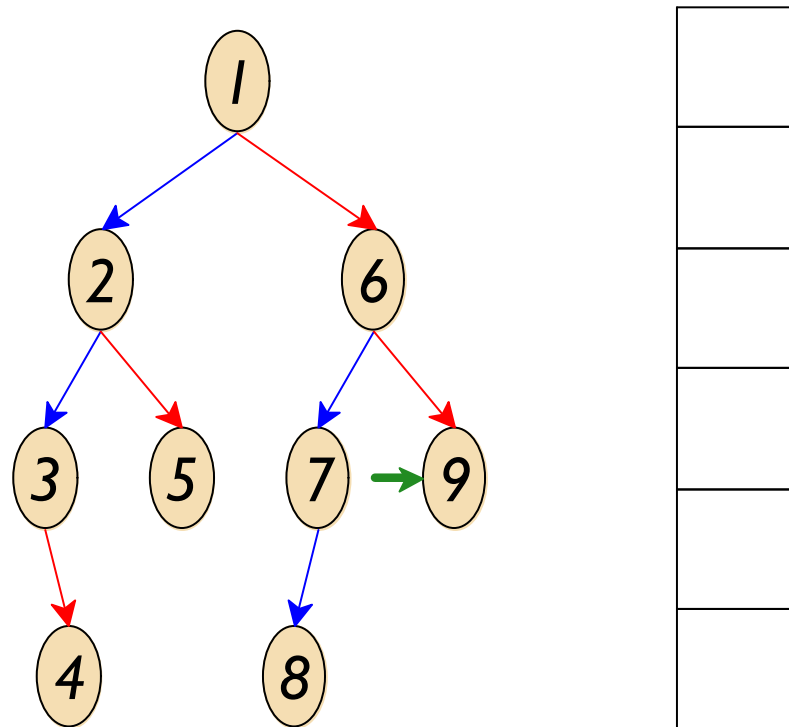
Depth-First Traversal (Preorder) via Stack



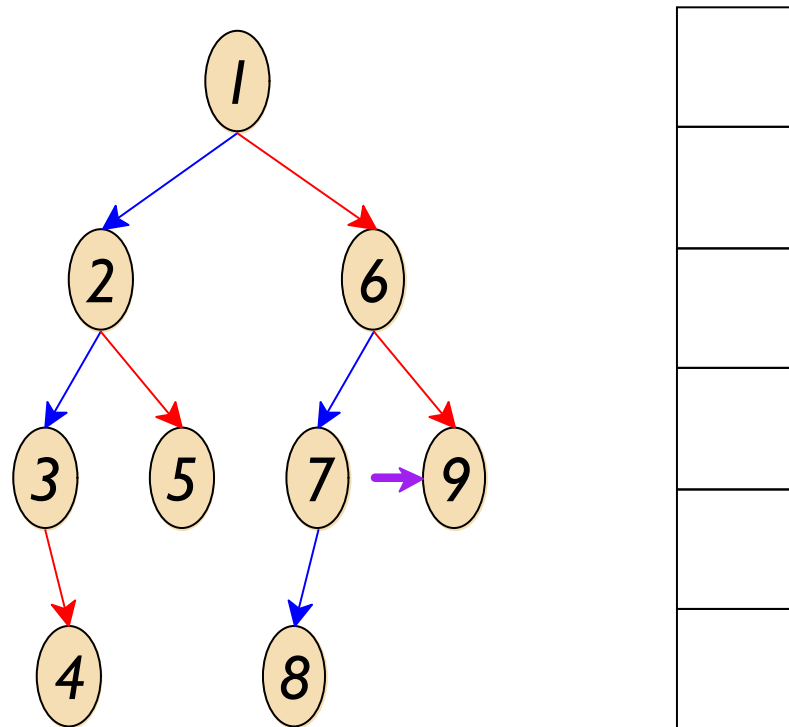
Depth-First Traversal (Preorder) via Stack



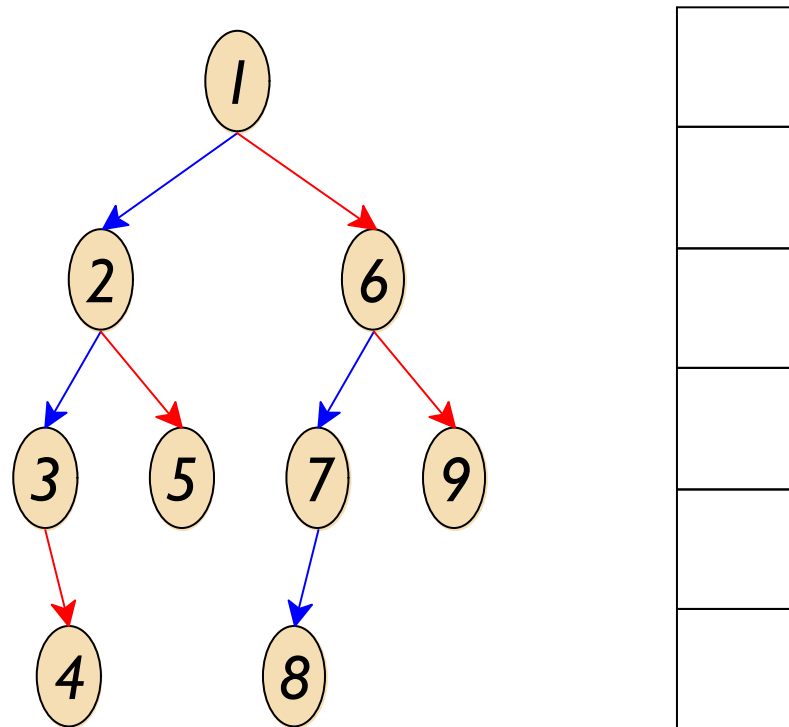
Depth-First Traversal (Preorder) via Stack



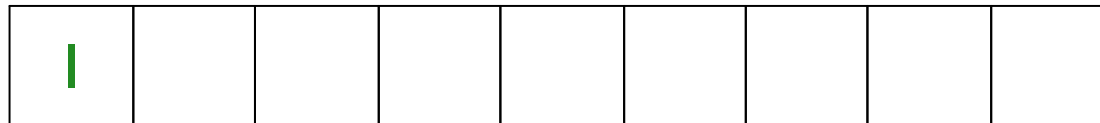
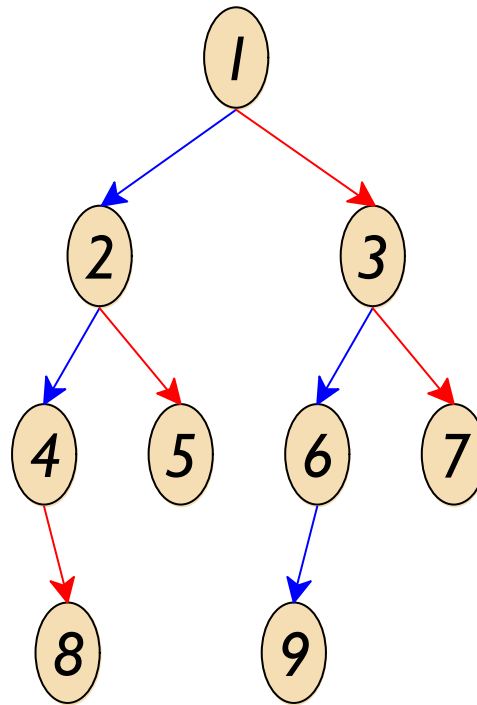
Depth-First Traversal (Preorder) via Stack



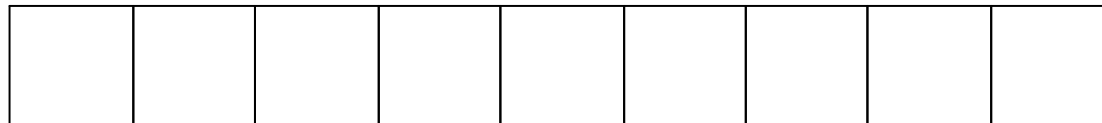
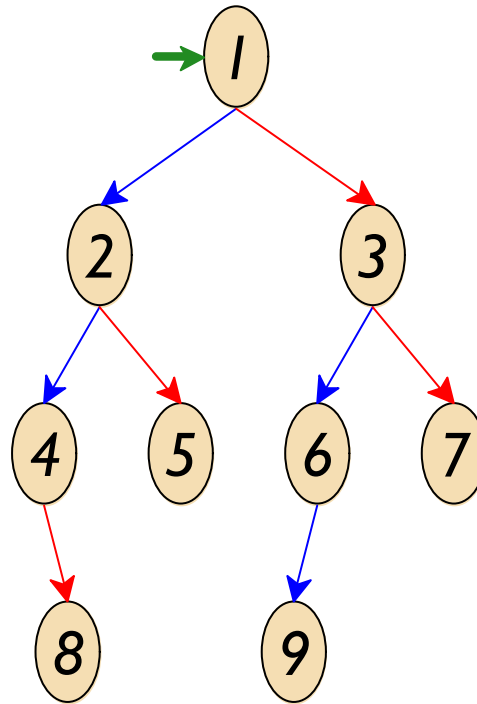
Depth-First Traversal (Preorder) via Stack



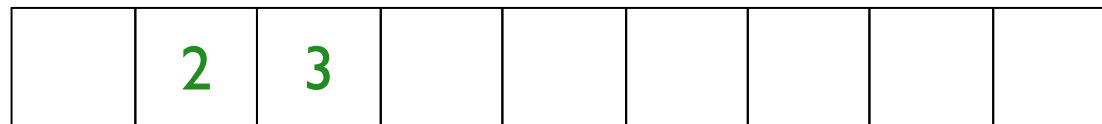
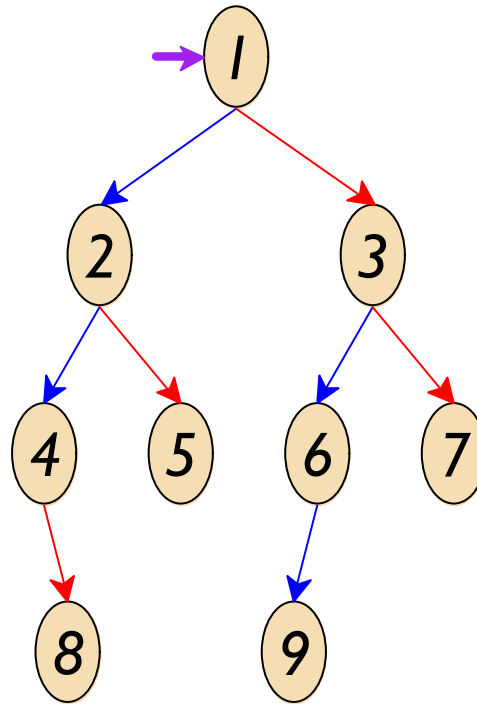
Breadth-First Traversal via Queue



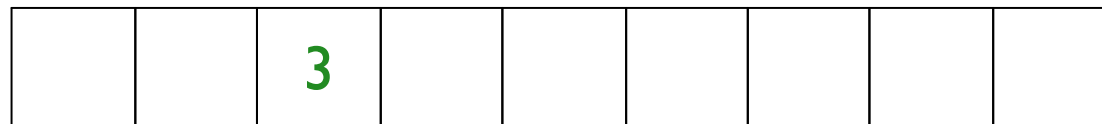
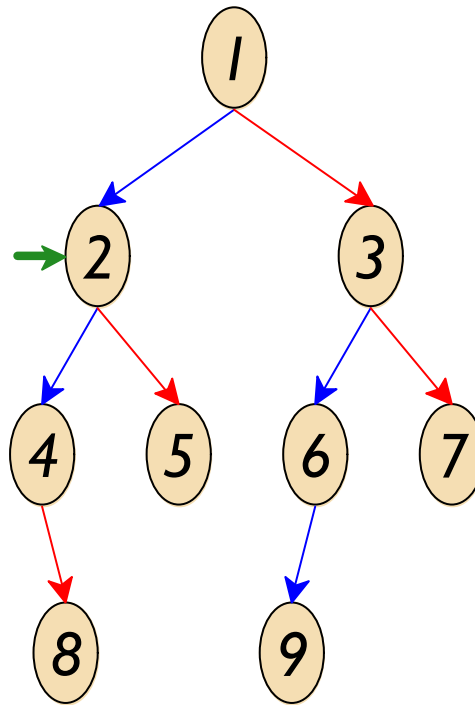
Breadth-First Traversal via Queue



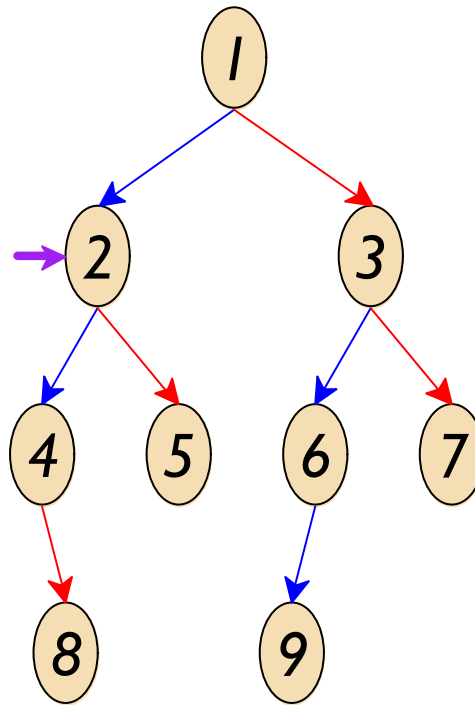
Breadth-First Traversal via Queue



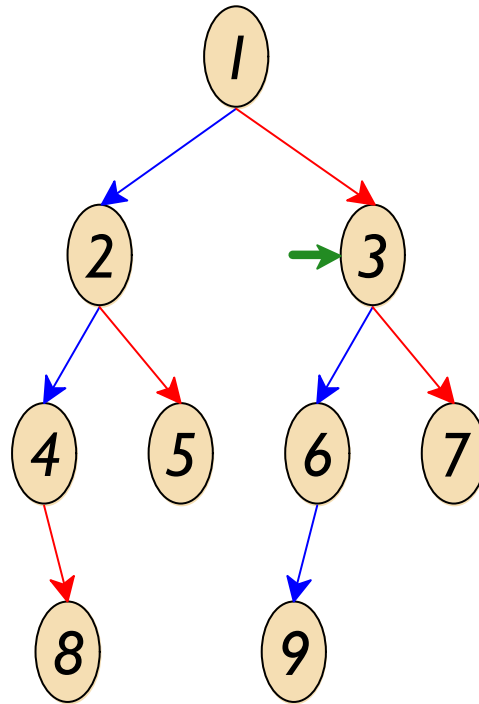
Breadth-First Traversal via Queue



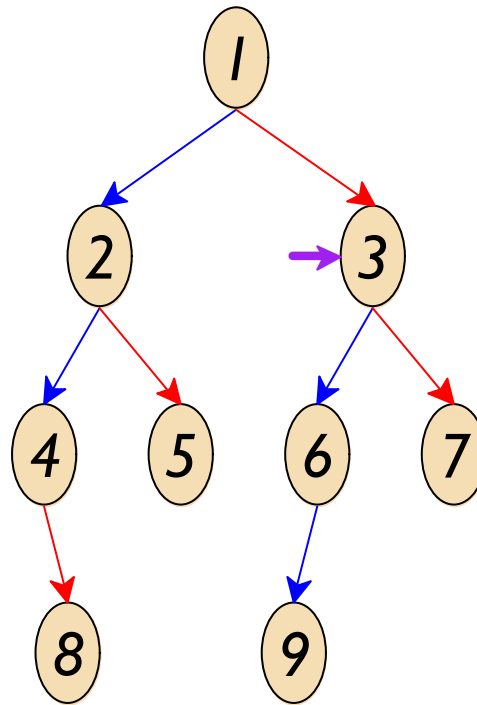
Breadth-First Traversal via Queue



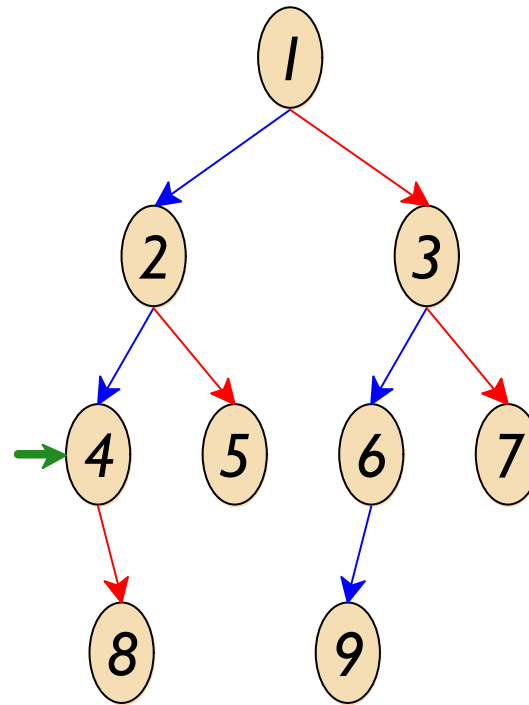
Breadth-First Traversal via Queue



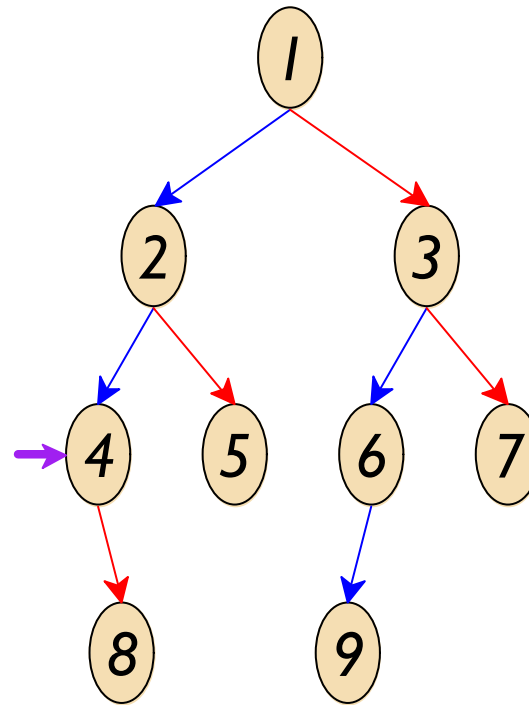
Breadth-First Traversal via Queue



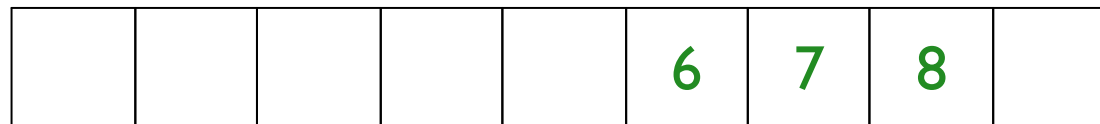
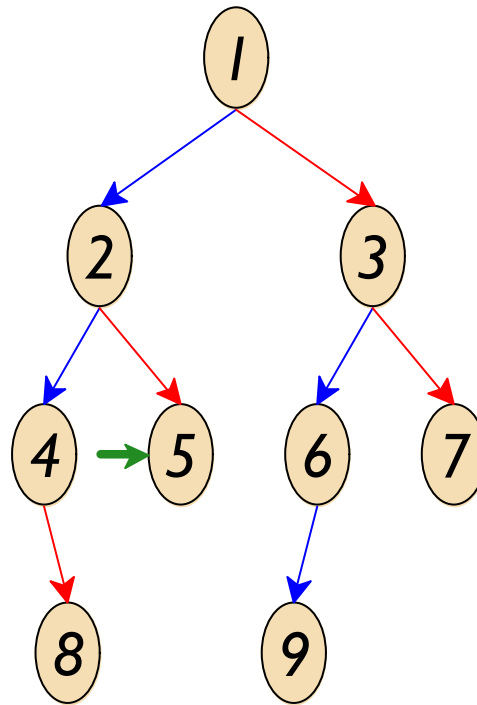
Breadth-First Traversal via Queue



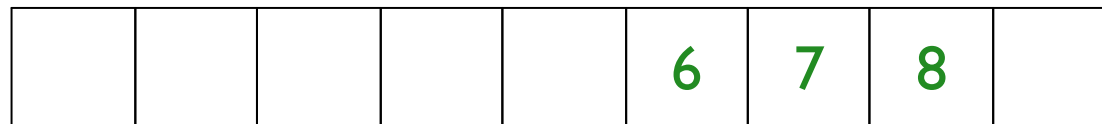
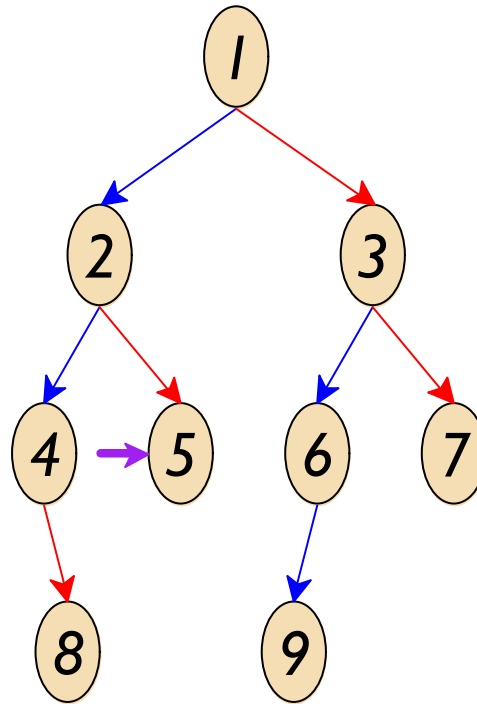
Breadth-First Traversal via Queue



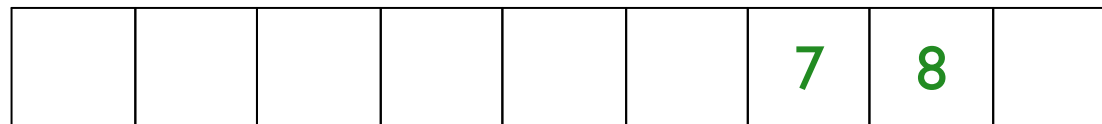
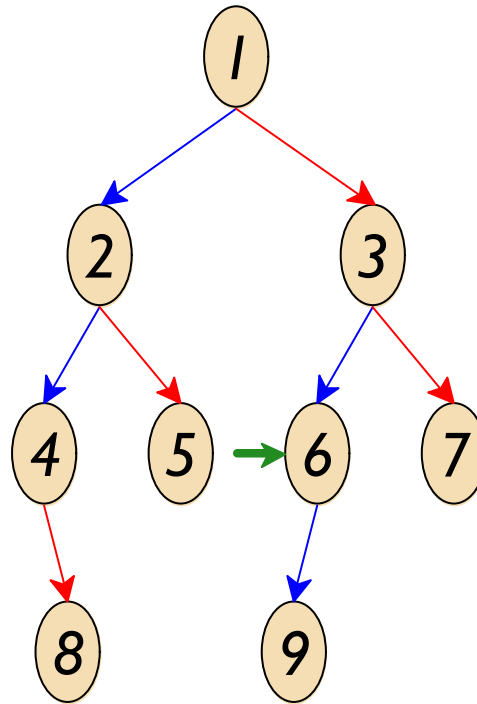
Breadth-First Traversal via Queue



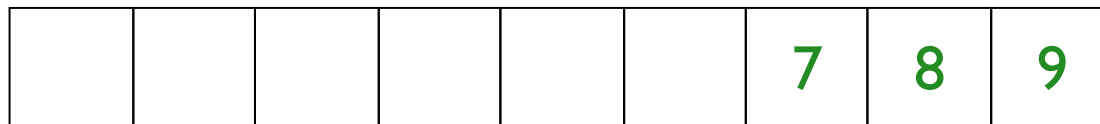
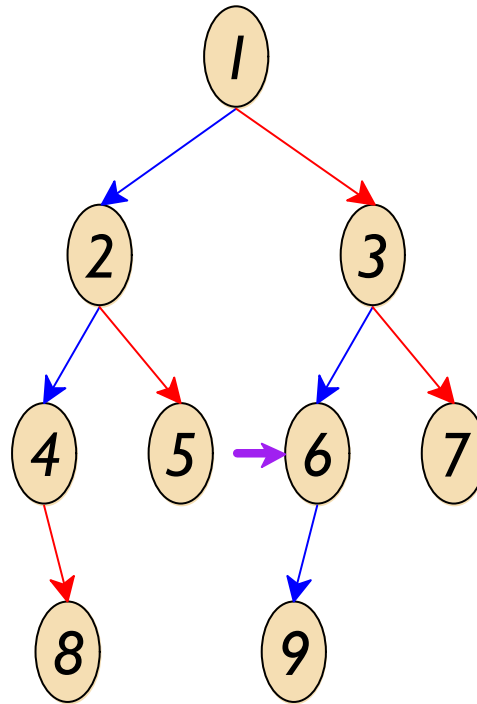
Breadth-First Traversal via Queue



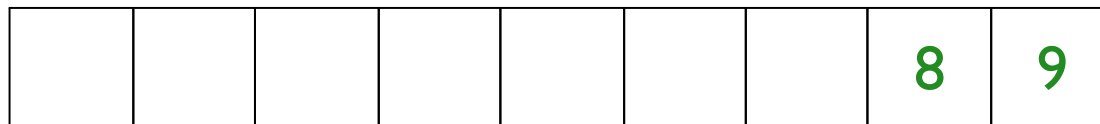
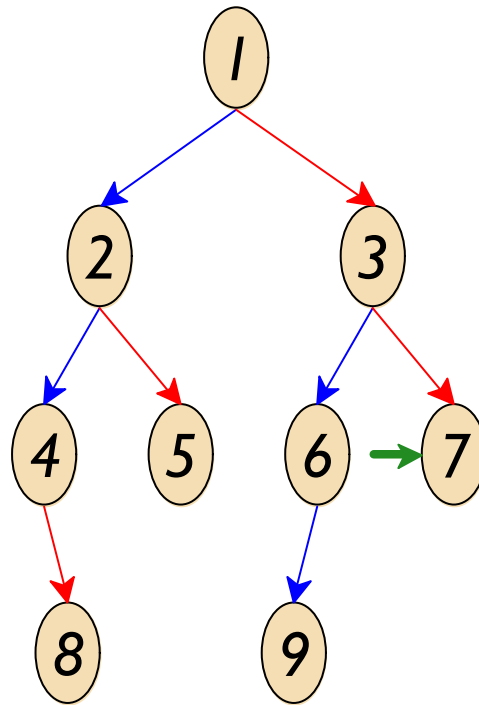
Breadth-First Traversal via Queue



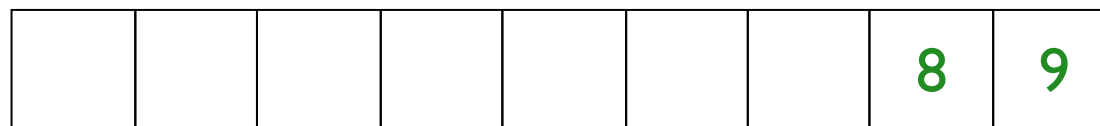
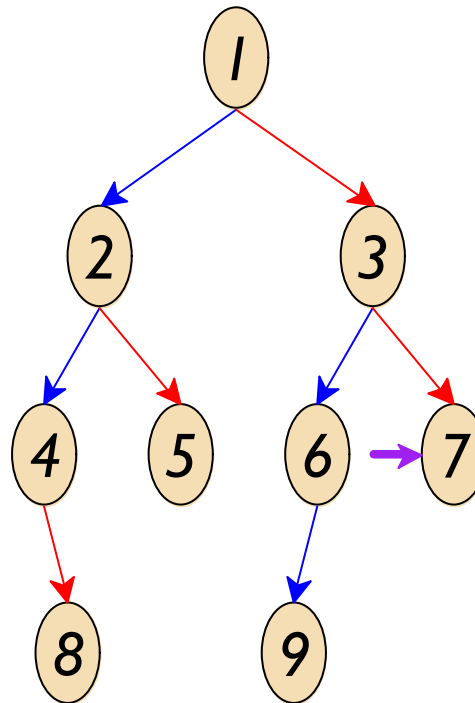
Breadth-First Traversal via Queue



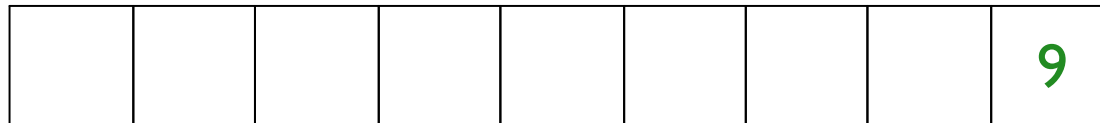
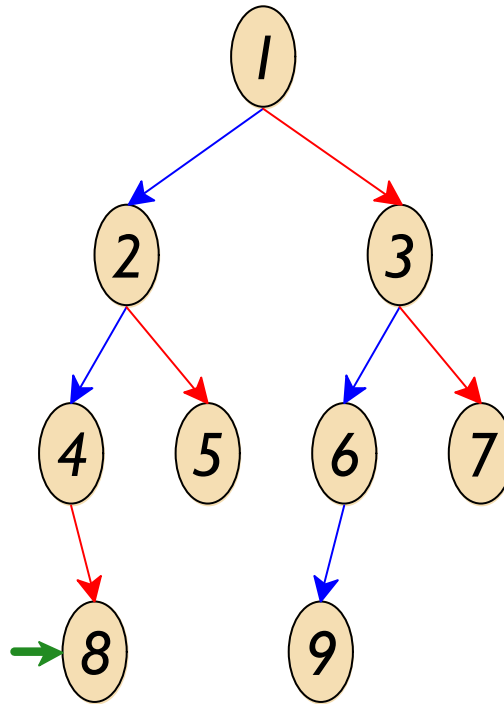
Breadth-First Traversal via Queue



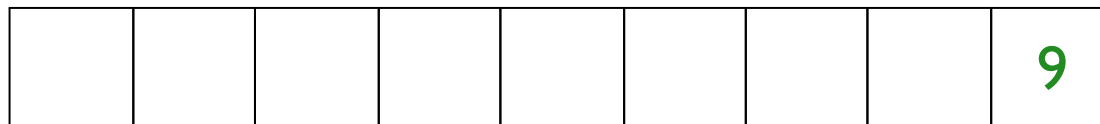
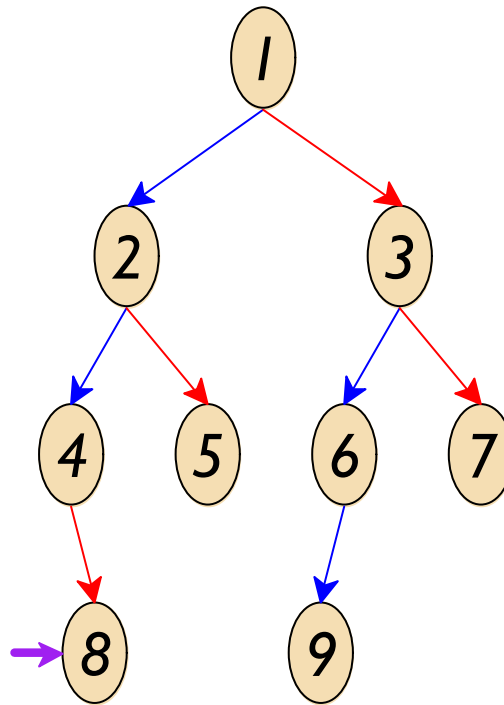
Breadth-First Traversal via Queue



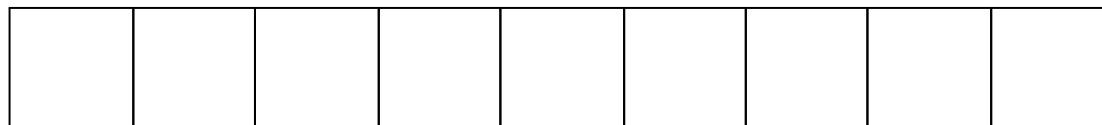
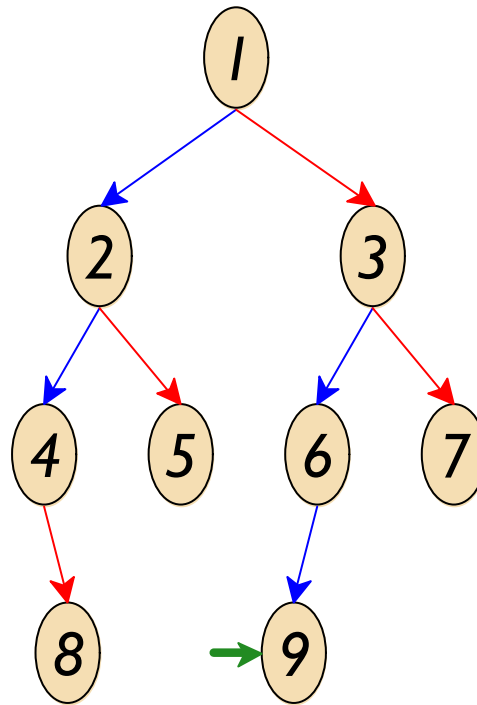
Breadth-First Traversal via Queue



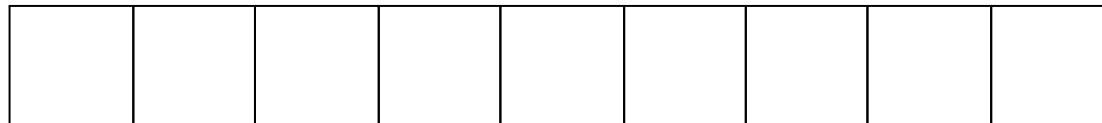
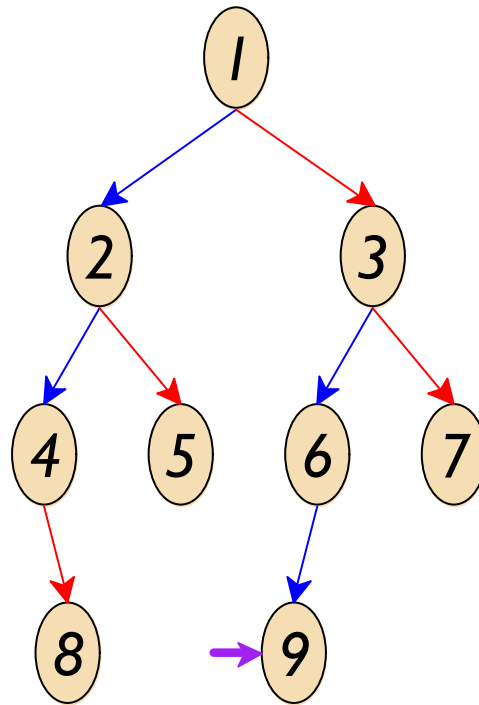
Breadth-First Traversal via Queue



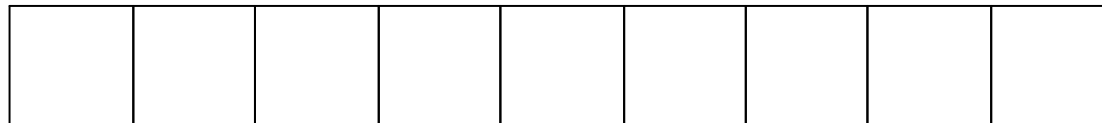
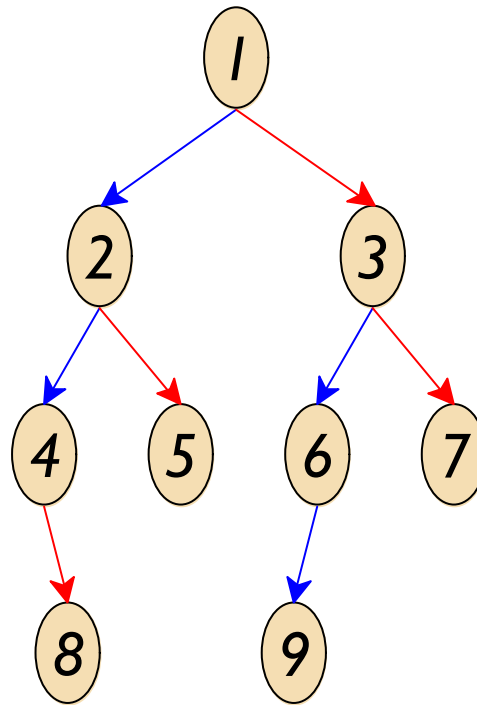
Breadth-First Traversal via Queue



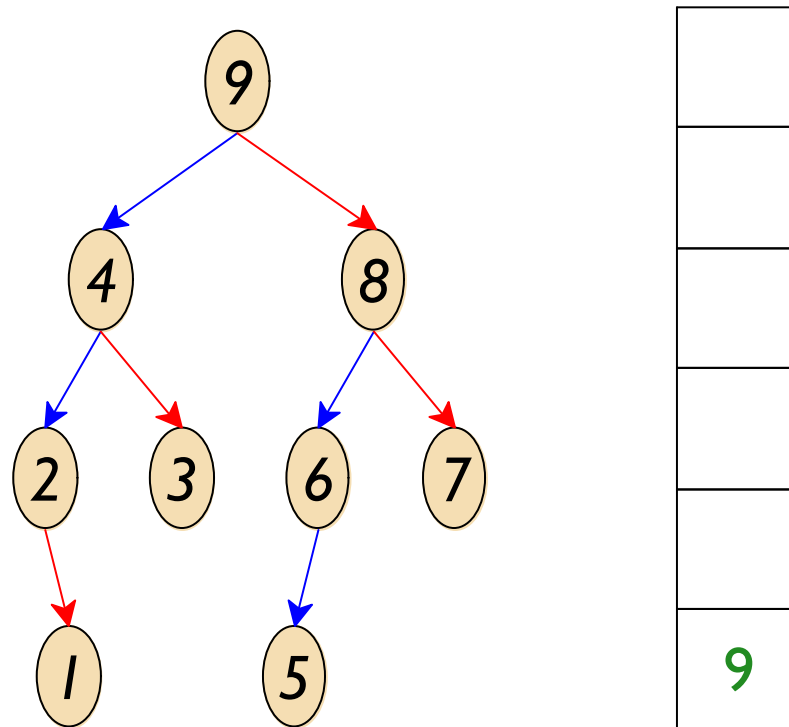
Breadth-First Traversal via Queue



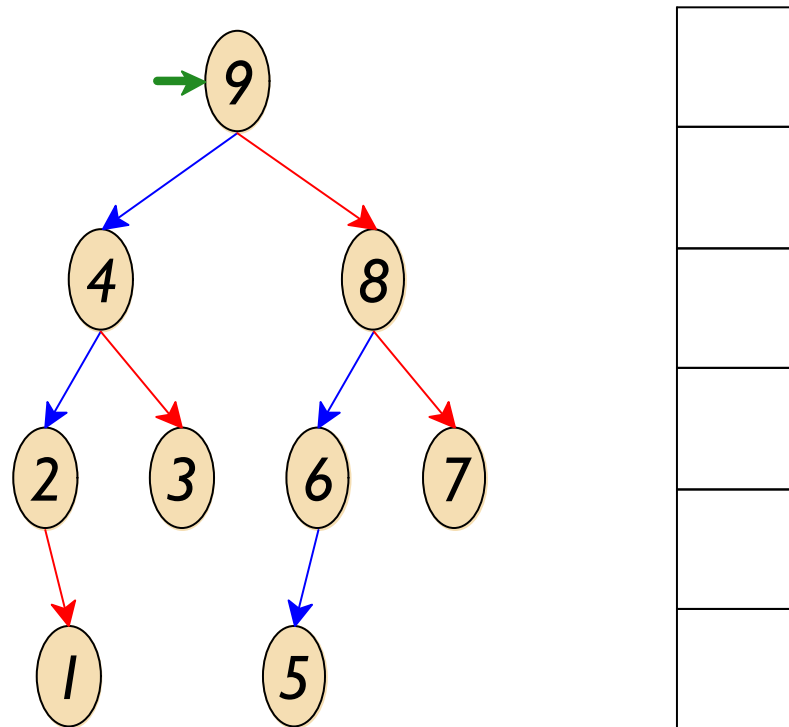
Breadth-First Traversal via Queue



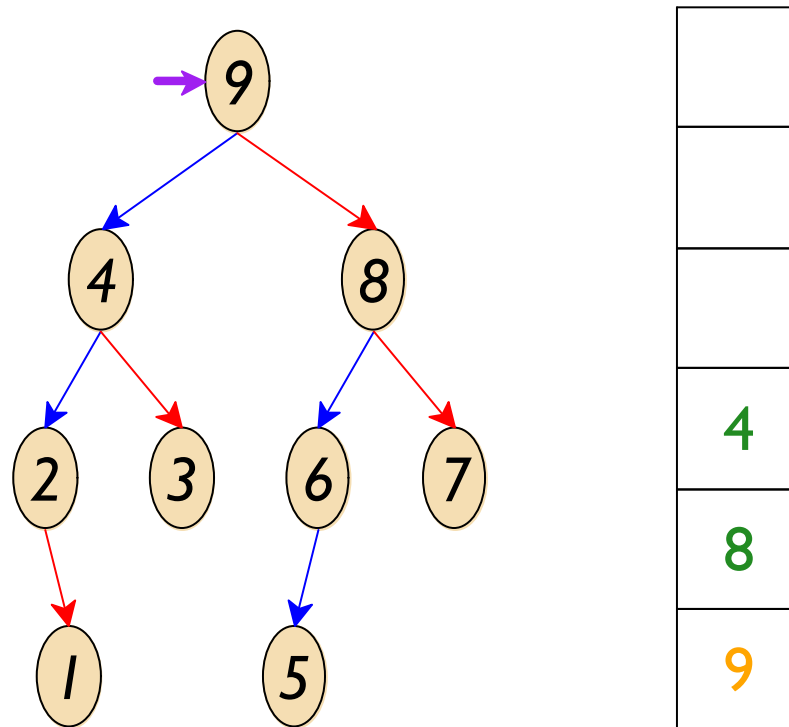
Depth-First Traversal (Postorder) via Stack



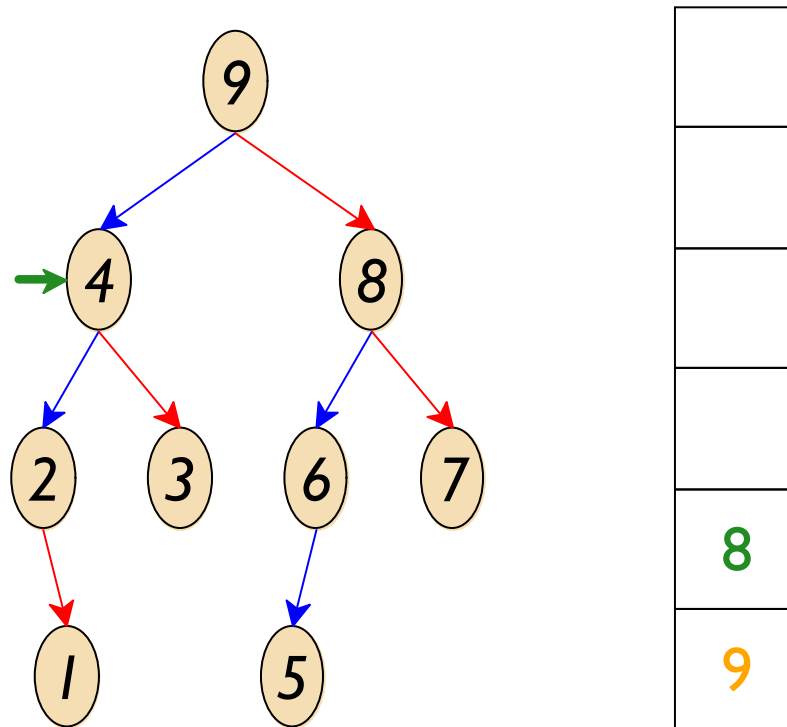
Depth-First Traversal (Postorder) via Stack



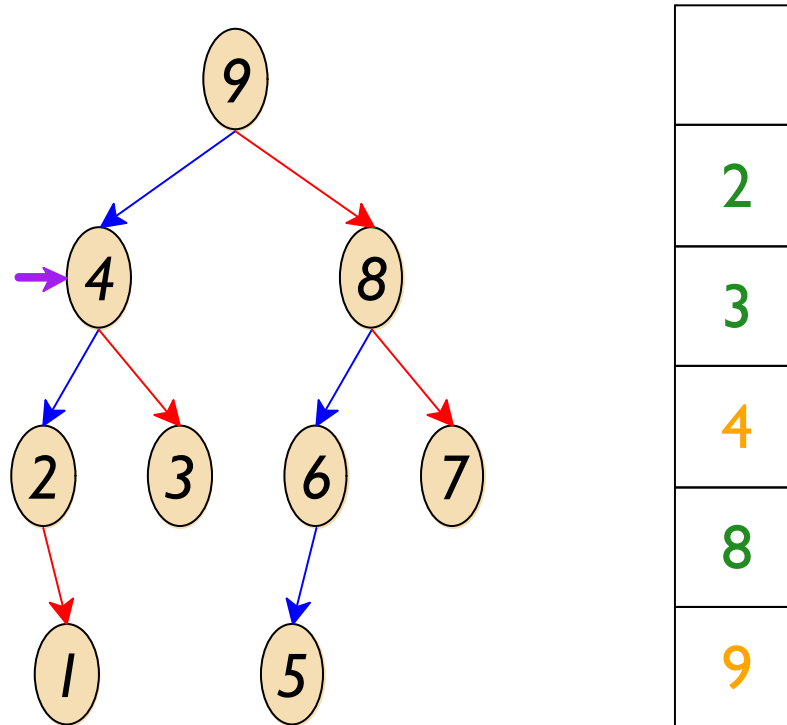
Depth-First Traversal (Postorder) via Stack



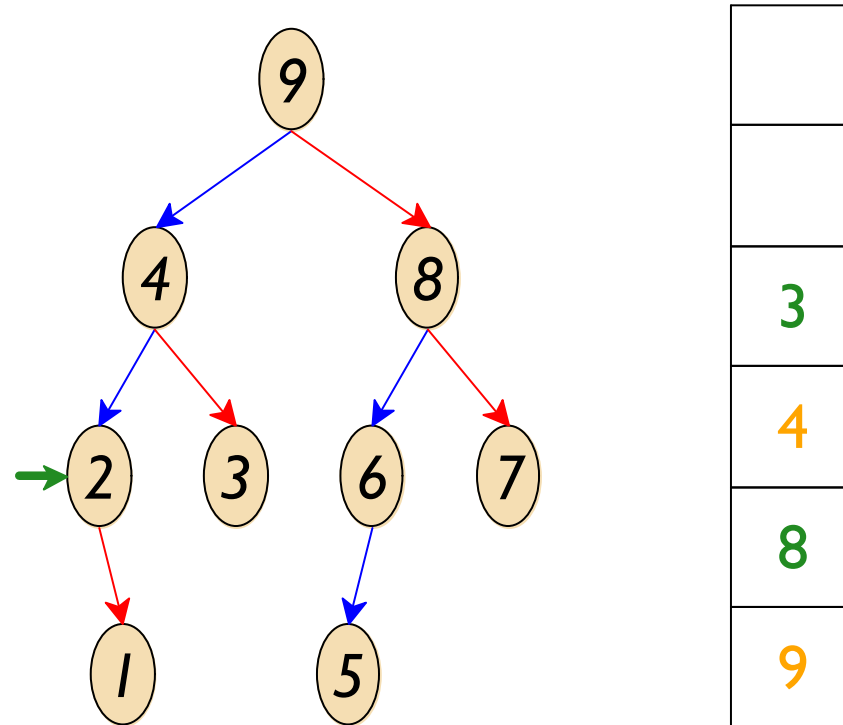
Depth-First Traversal (Postorder) via Stack



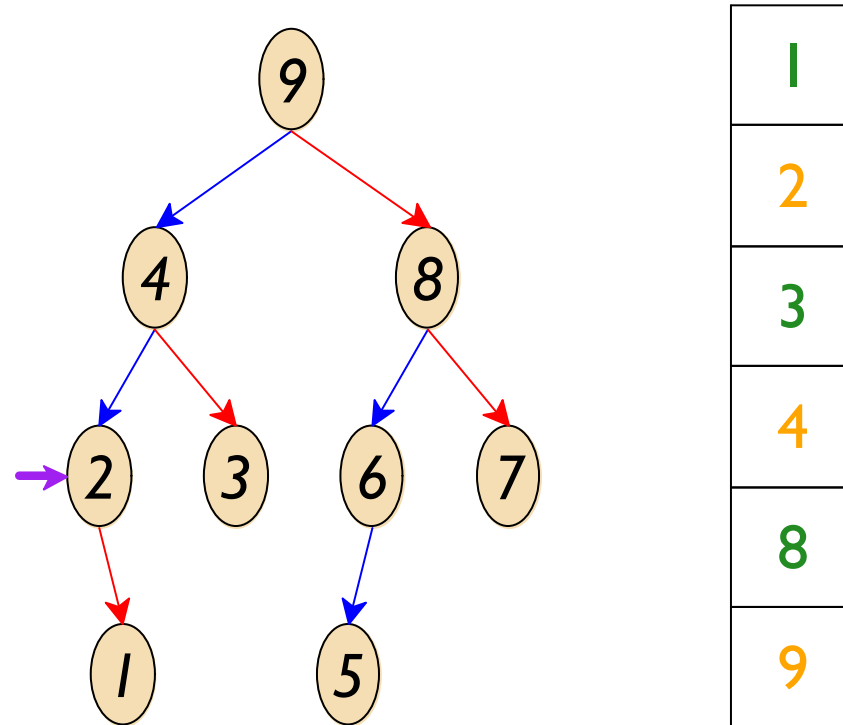
Depth-First Traversal (Postorder) via Stack



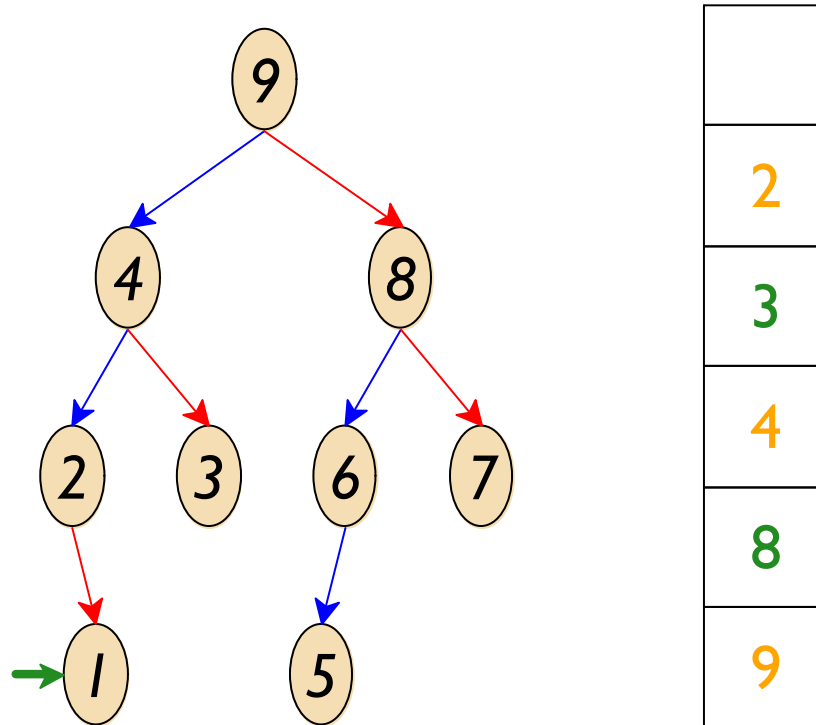
Depth-First Traversal (Postorder) via Stack



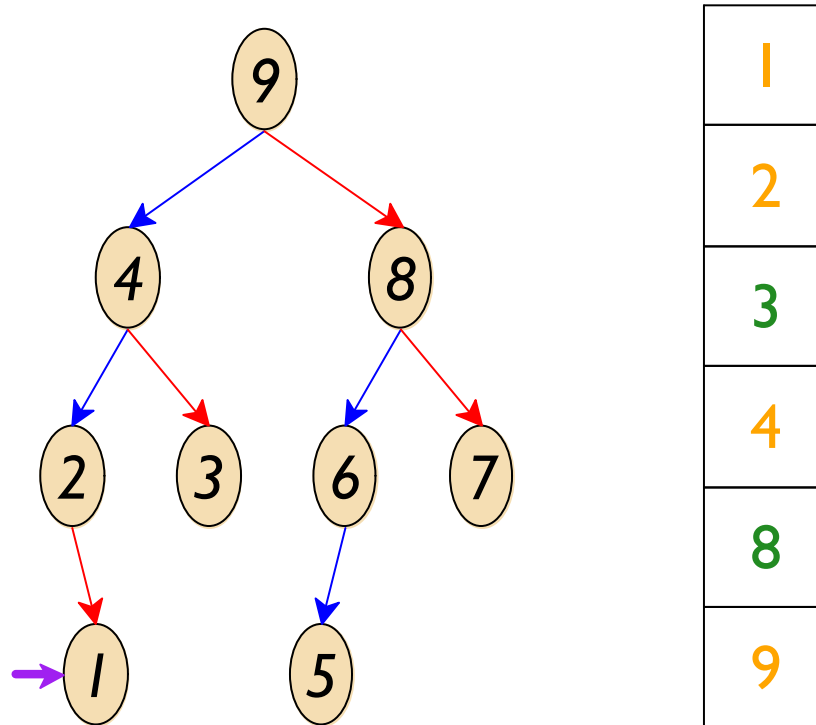
Depth-First Traversal (Postorder) via Stack



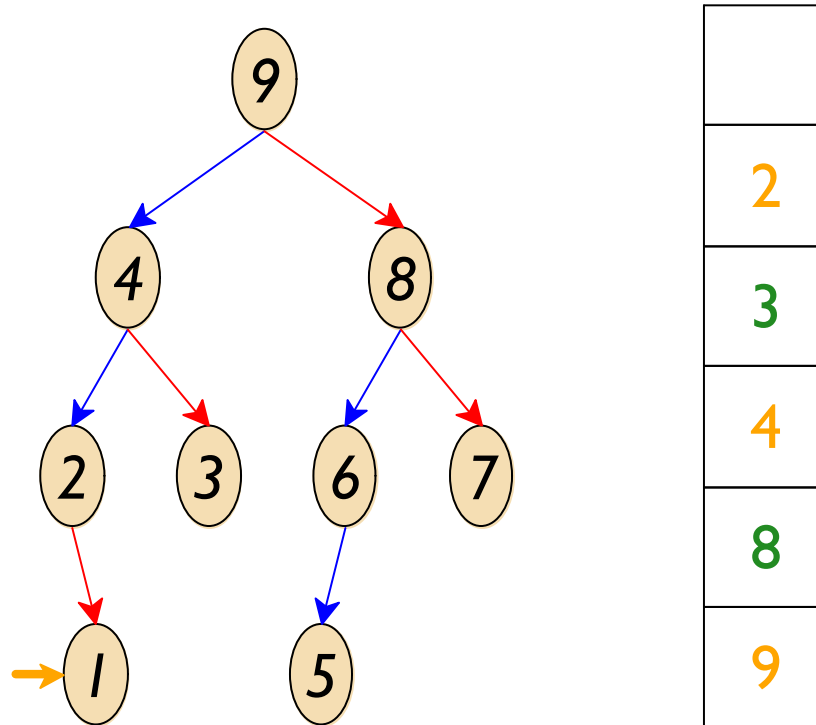
Depth-First Traversal (Postorder) via Stack



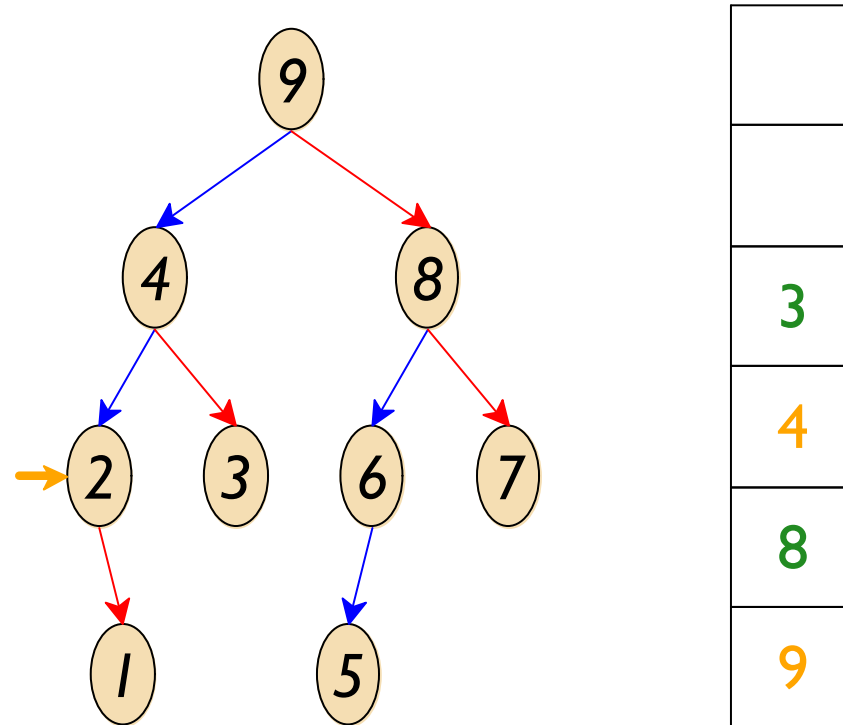
Depth-First Traversal (Postorder) via Stack



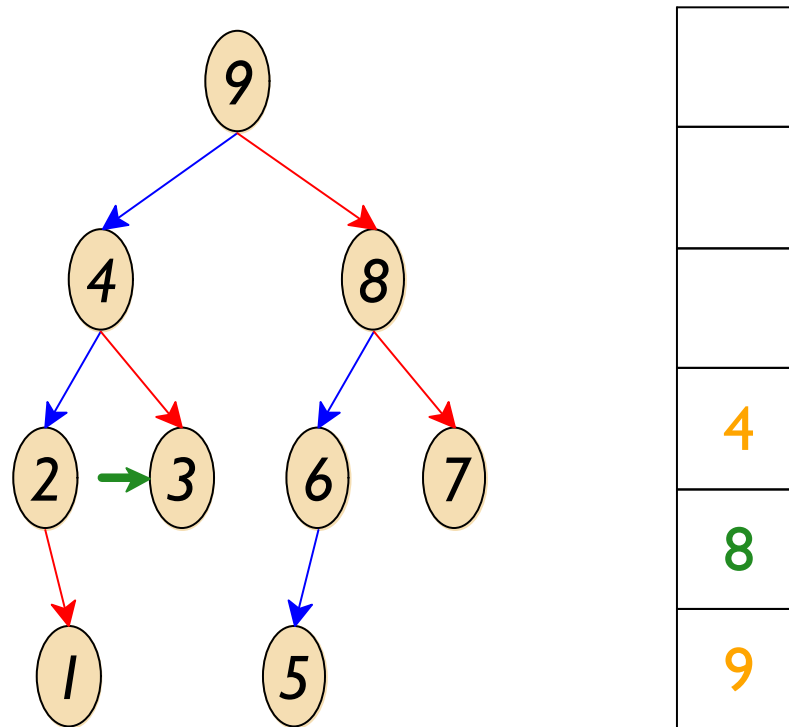
Depth-First Traversal (Postorder) via Stack



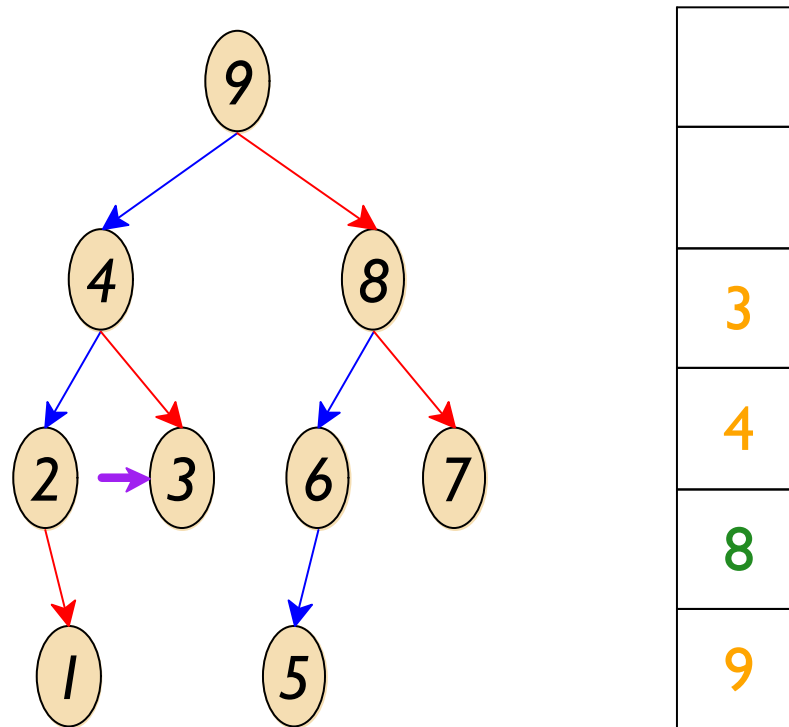
Depth-First Traversal (Postorder) via Stack



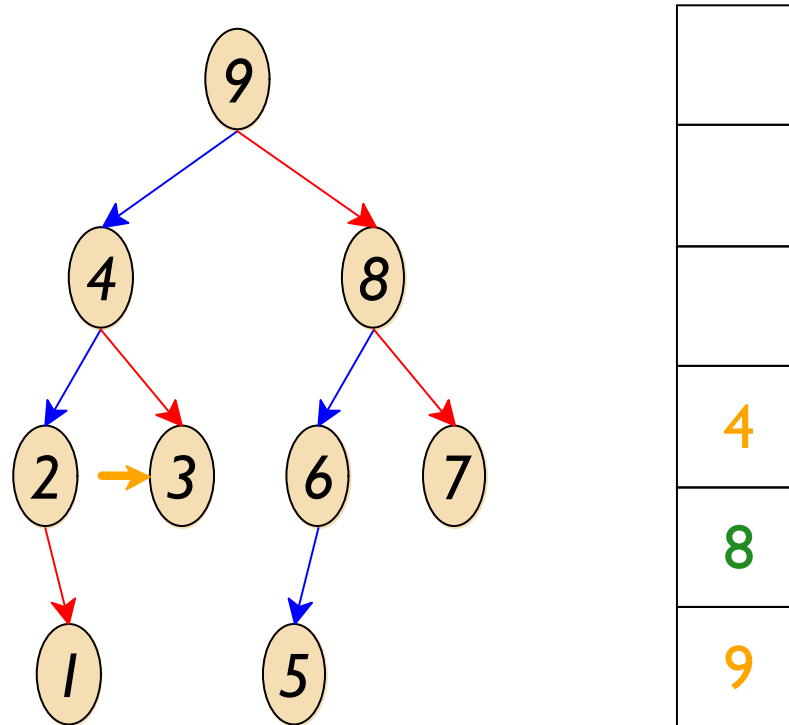
Depth-First Traversal (Postorder) via Stack



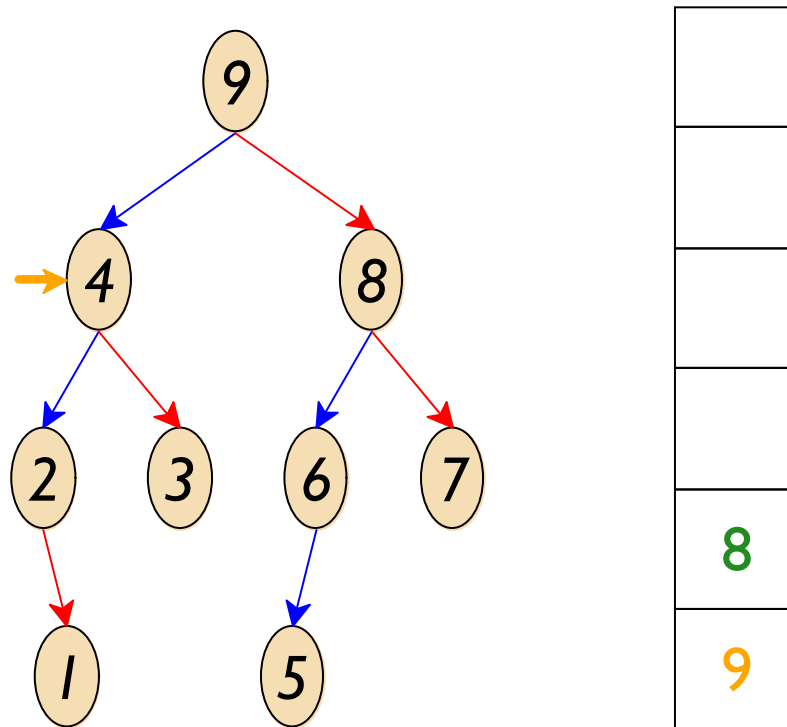
Depth-First Traversal (Postorder) via Stack



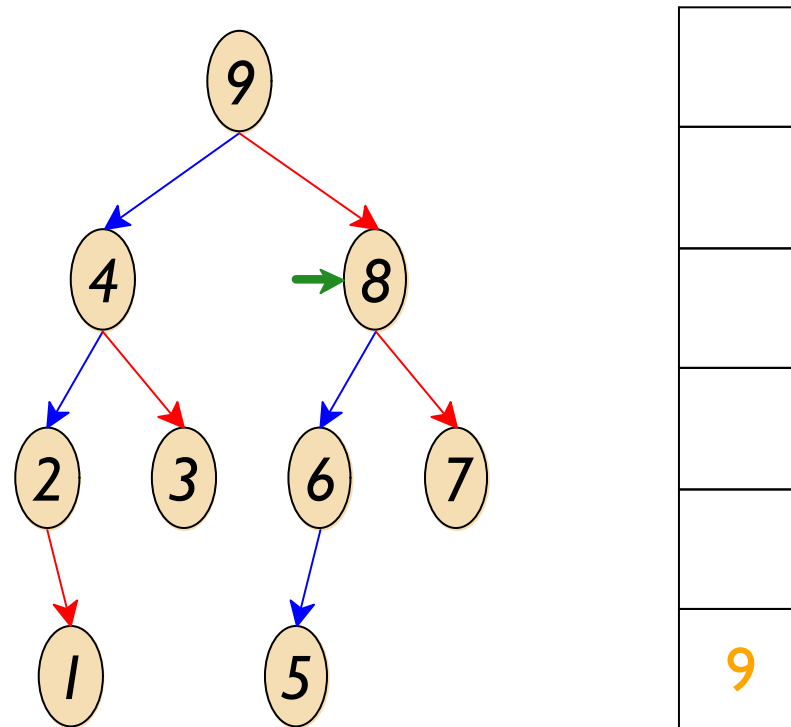
Depth-First Traversal (Postorder) via Stack



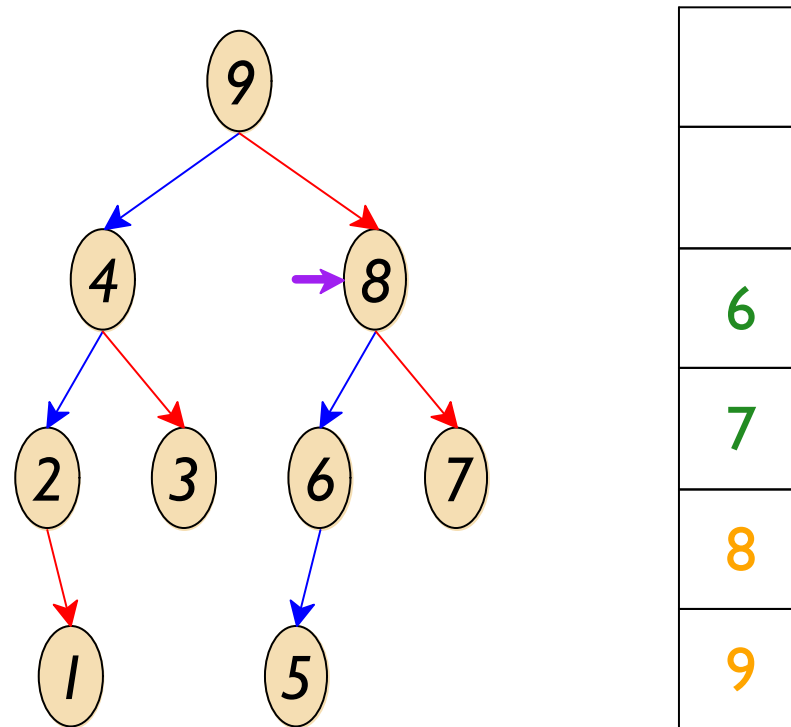
Depth-First Traversal (Postorder) via Stack



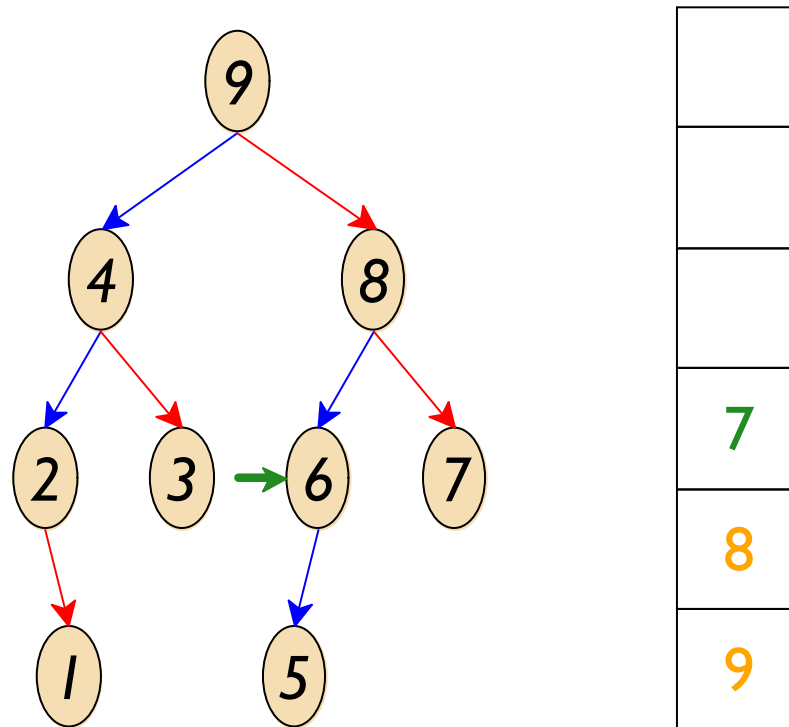
Depth-First Traversal (Postorder) via Stack



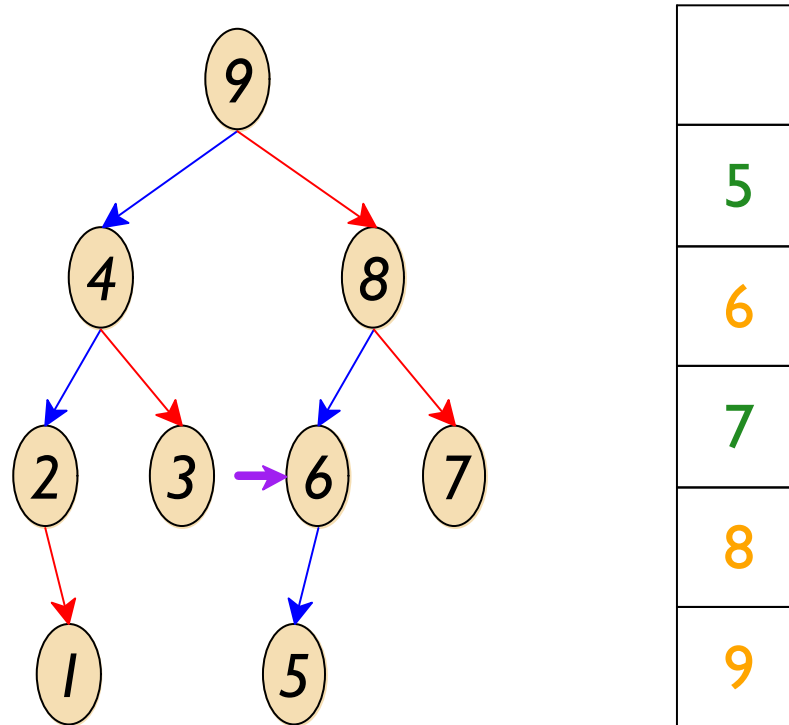
Depth-First Traversal (Postorder) via Stack



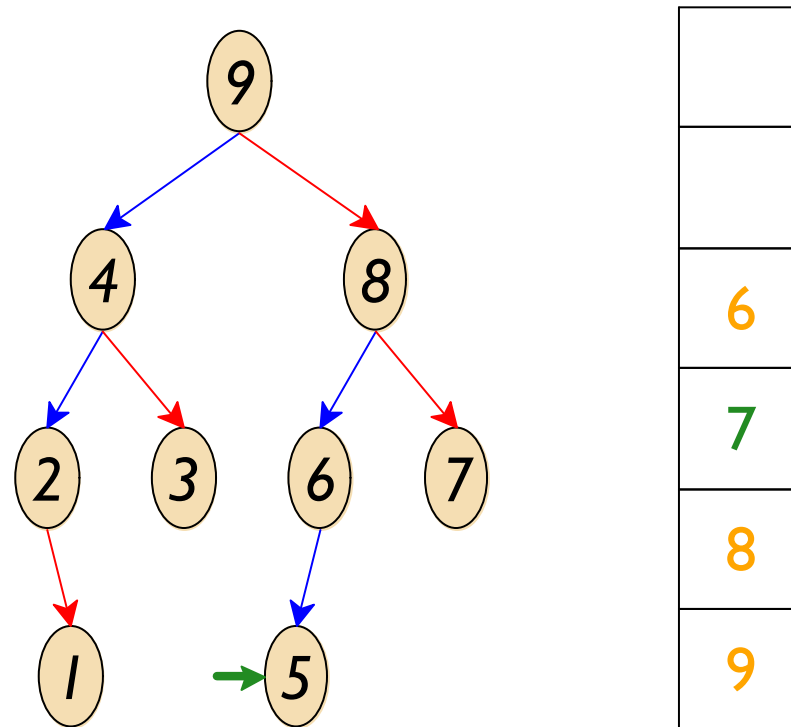
Depth-First Traversal (Postorder) via Stack



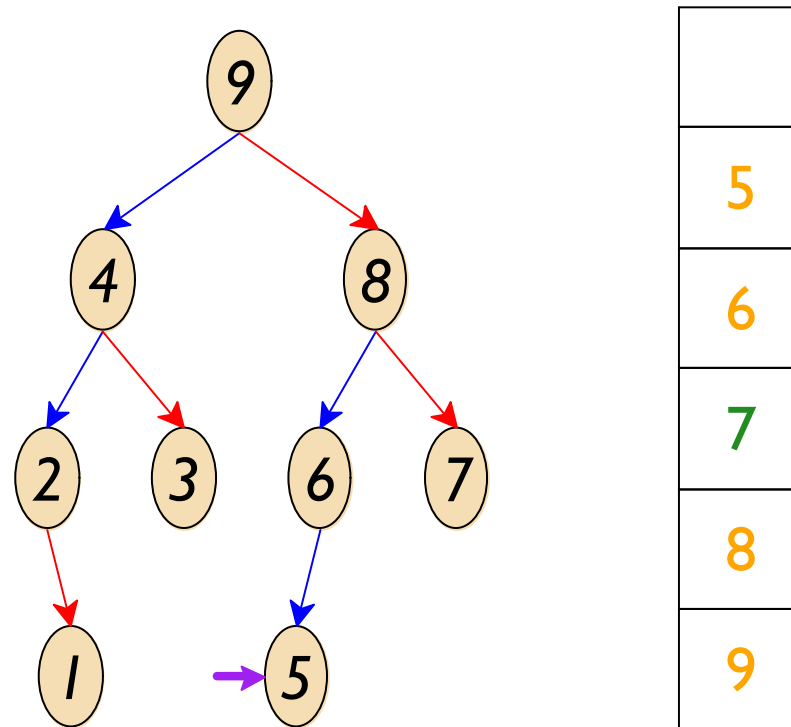
Depth-First Traversal (Postorder) via Stack



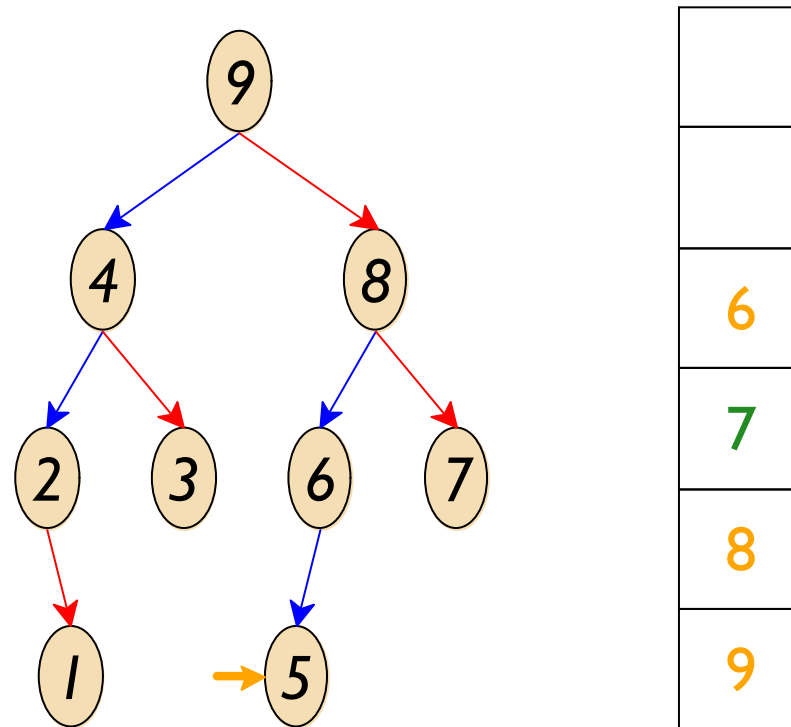
Depth-First Traversal (Postorder) via Stack



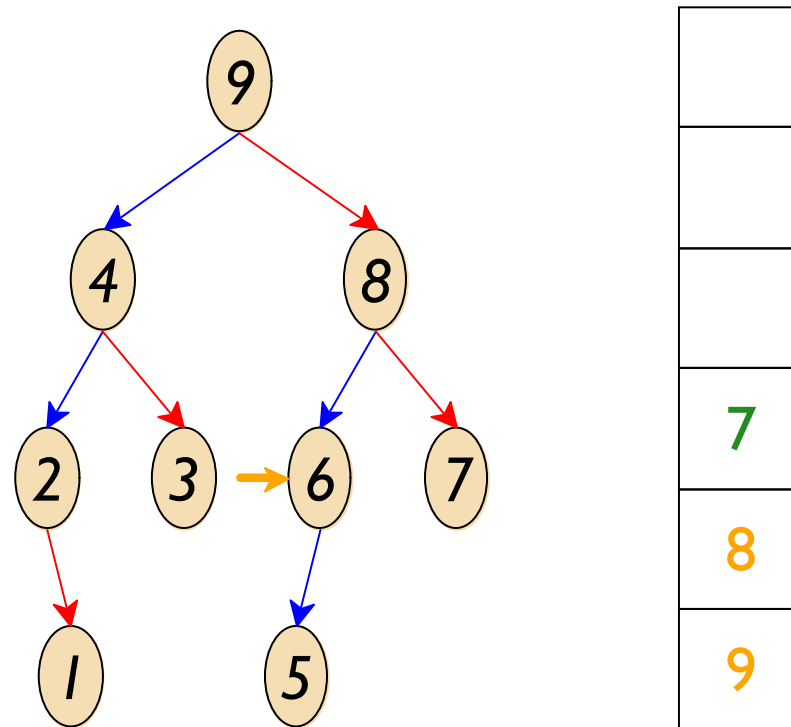
Depth-First Traversal (Postorder) via Stack



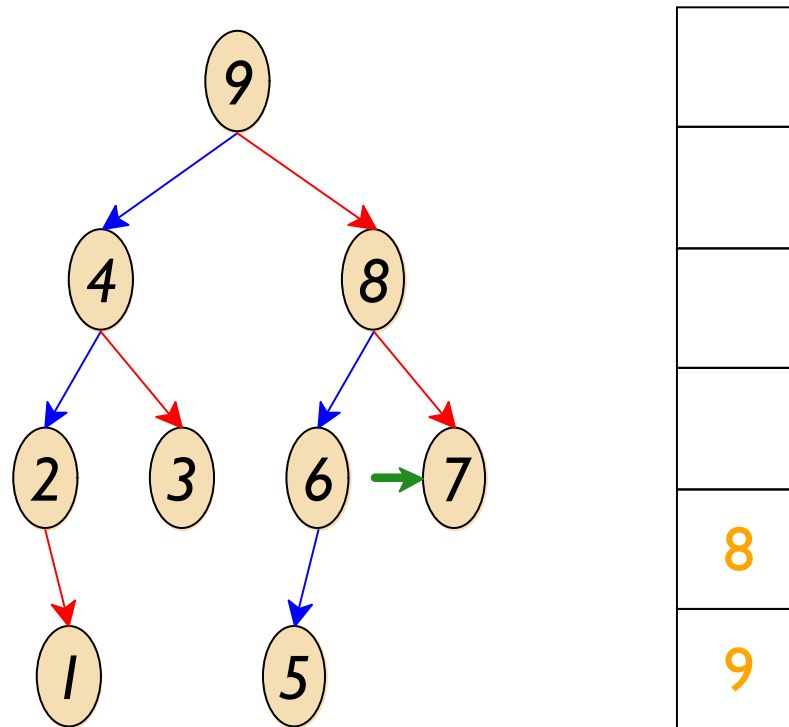
Depth-First Traversal (Postorder) via Stack



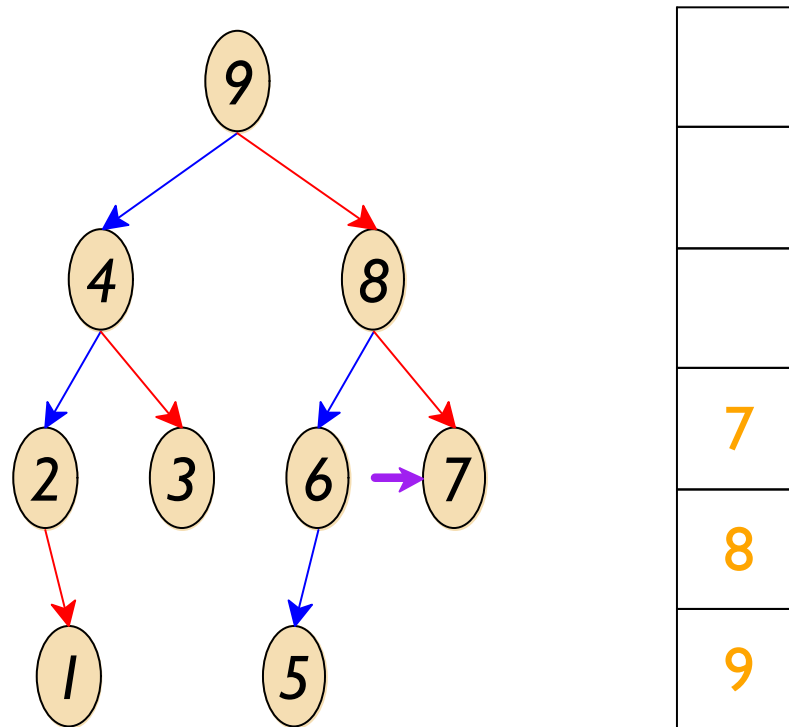
Depth-First Traversal (Postorder) via Stack



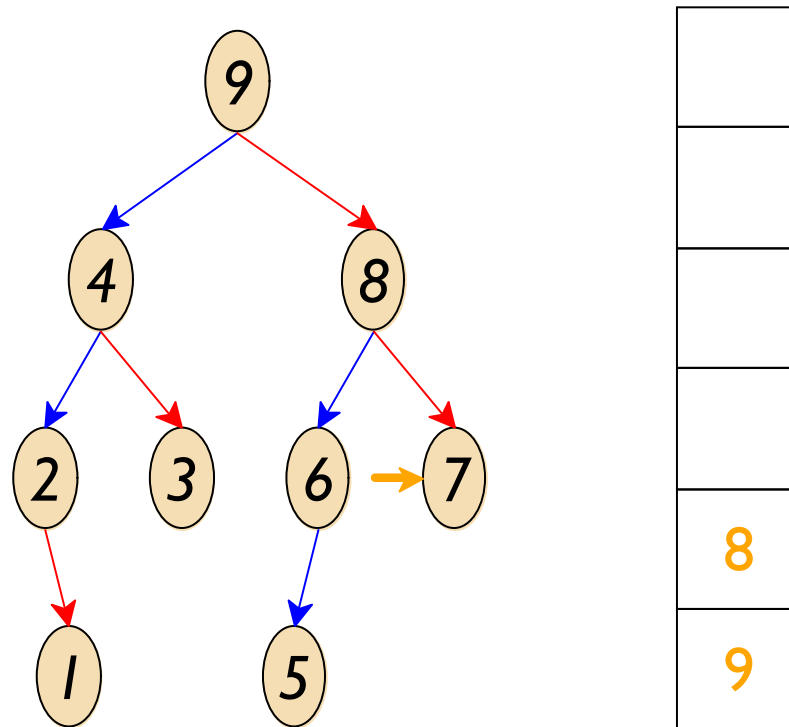
Depth-First Traversal (Postorder) via Stack



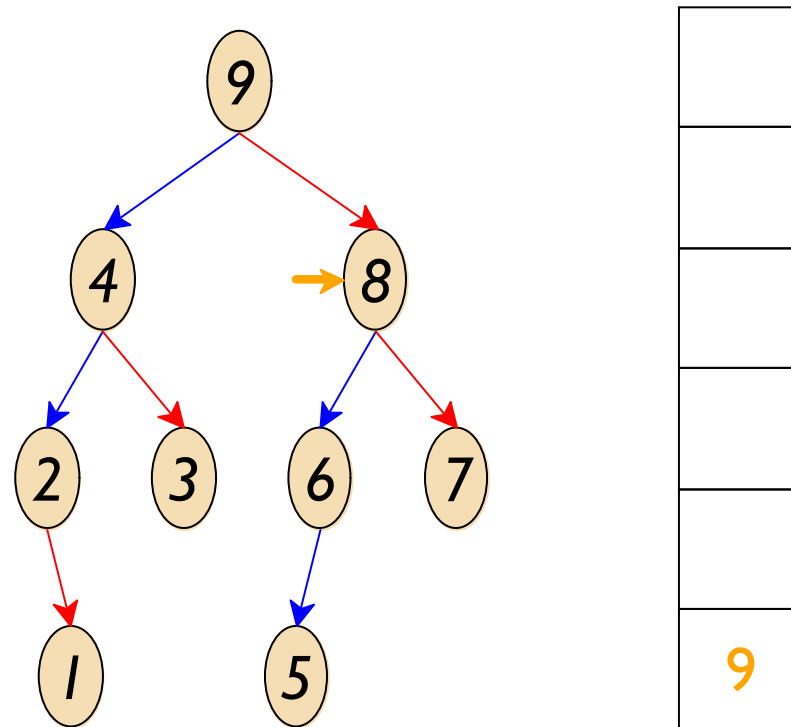
Depth-First Traversal (Postorder) via Stack



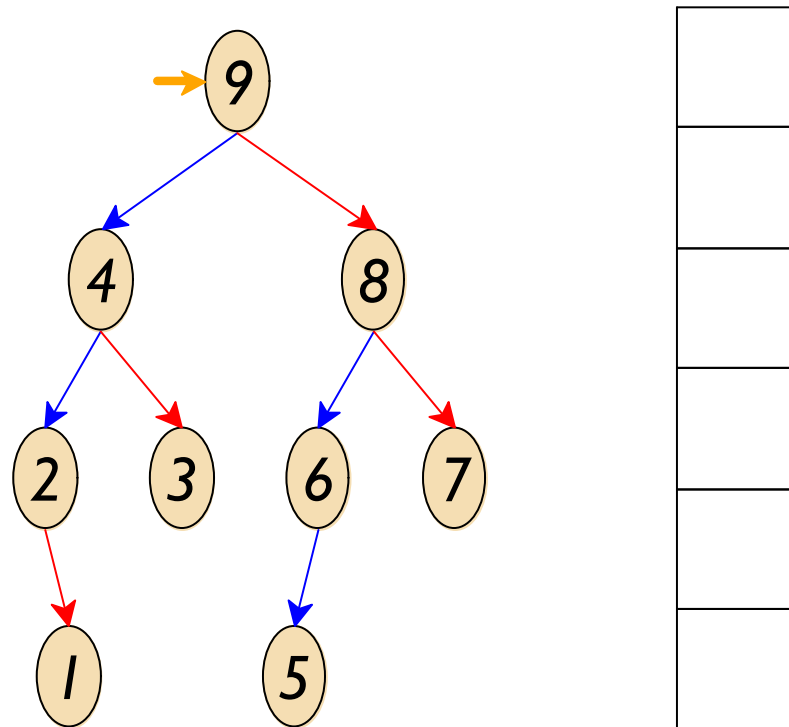
Depth-First Traversal (Postorder) via Stack



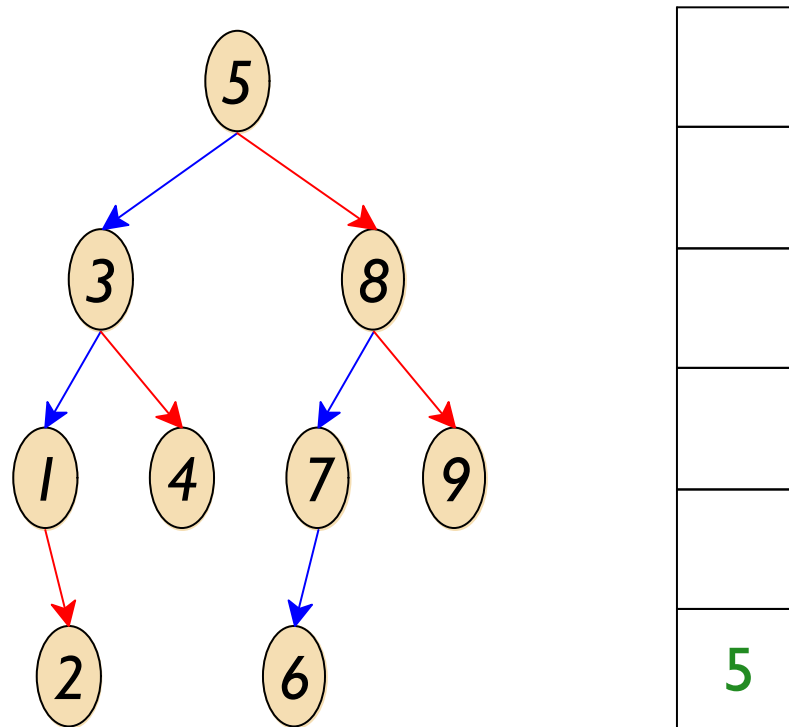
Depth-First Traversal (Postorder) via Stack



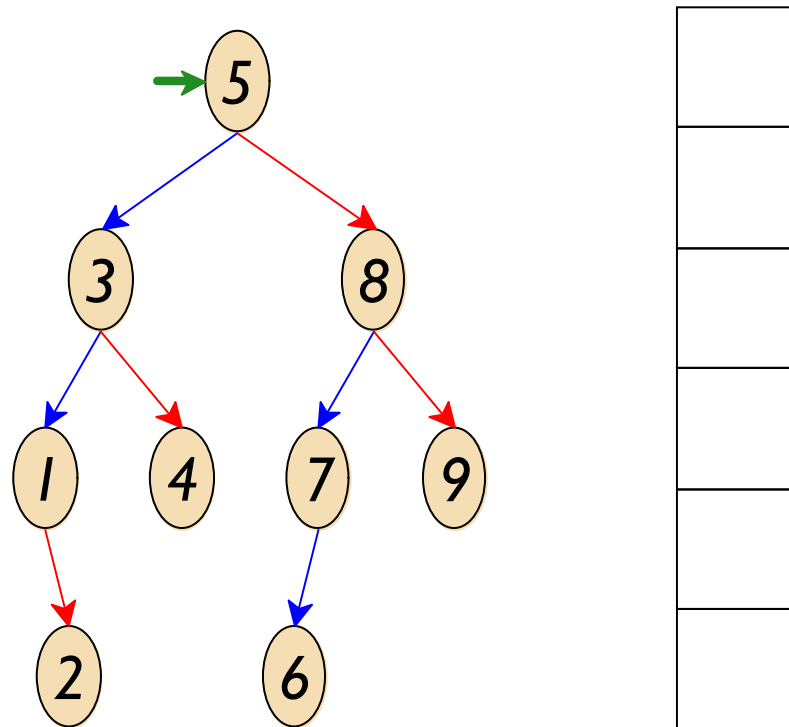
Depth-First Traversal (Postorder) via Stack



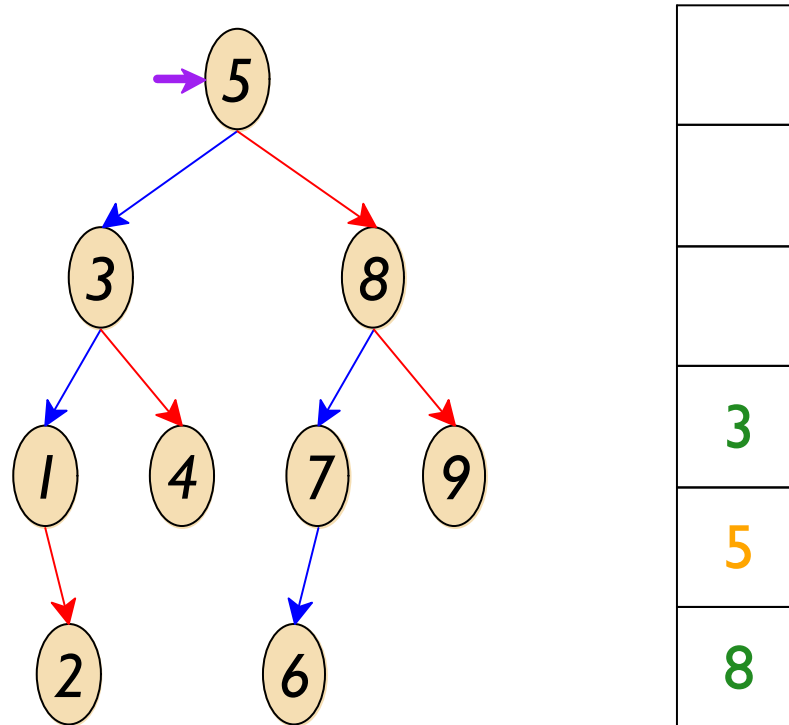
Depth-First Traversal (Inorder) via Stack



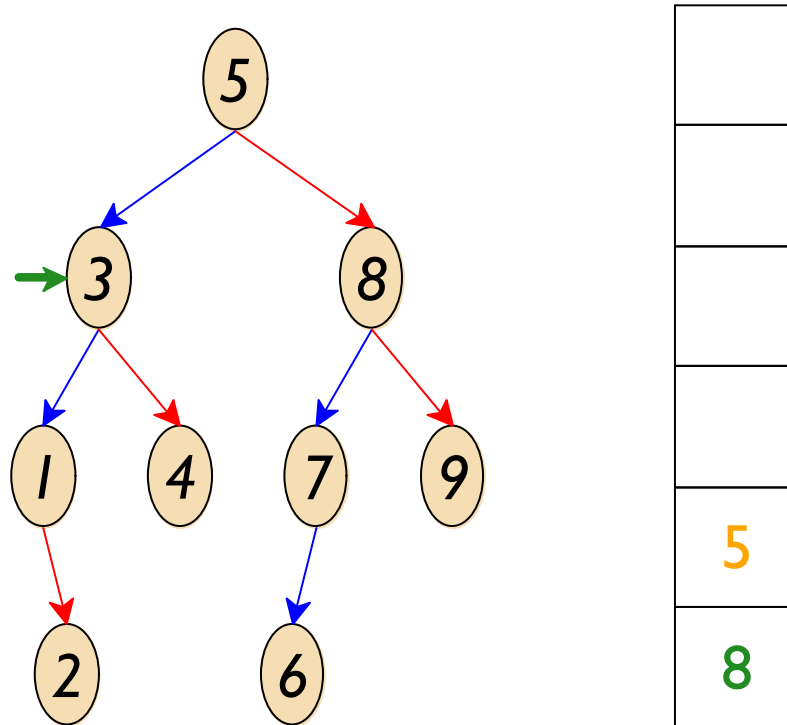
Depth-First Traversal (Inorder) via Stack



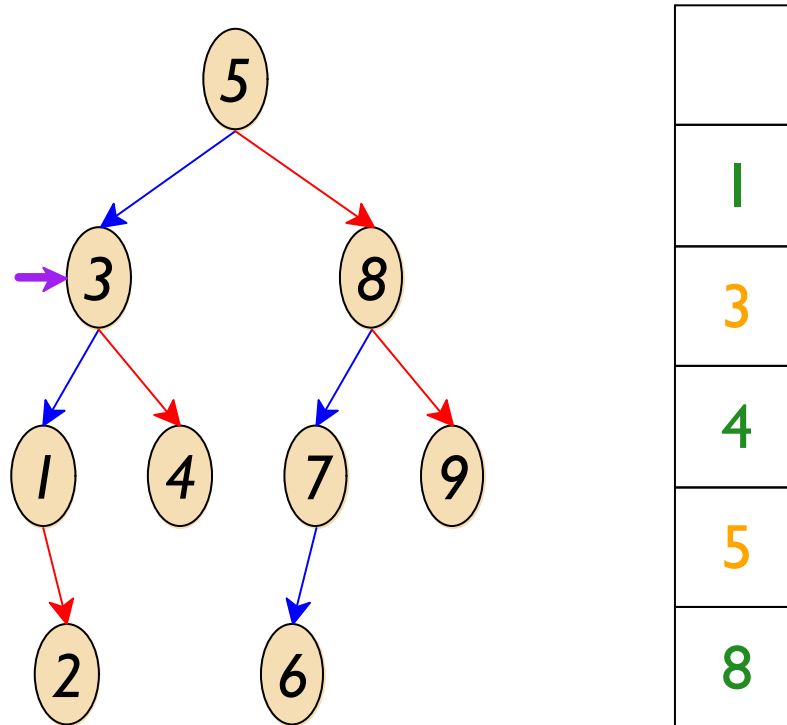
Depth-First Traversal (Inorder) via Stack



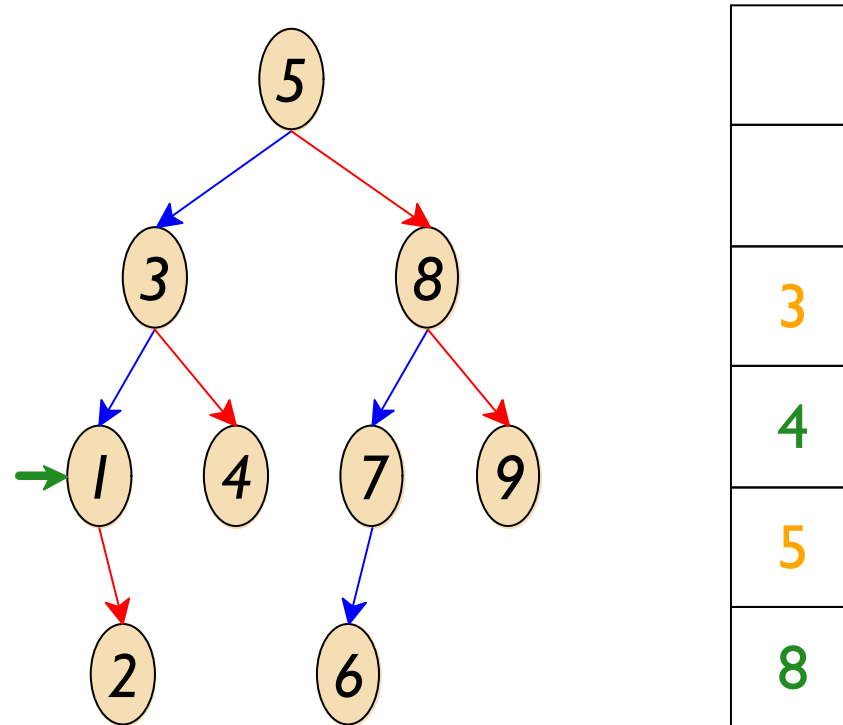
Depth-First Traversal (Inorder) via Stack



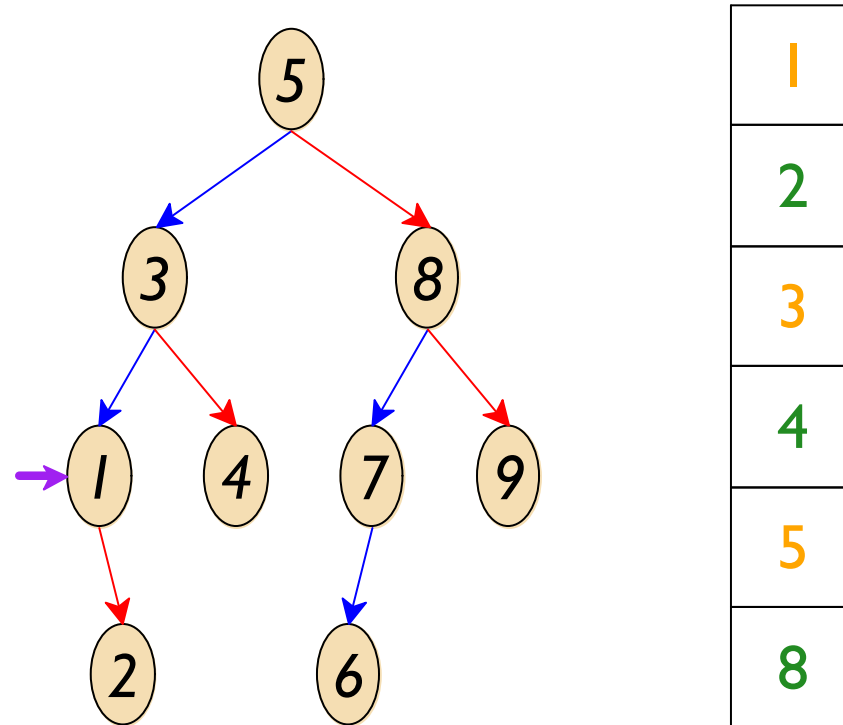
Depth-First Traversal (Inorder) via Stack



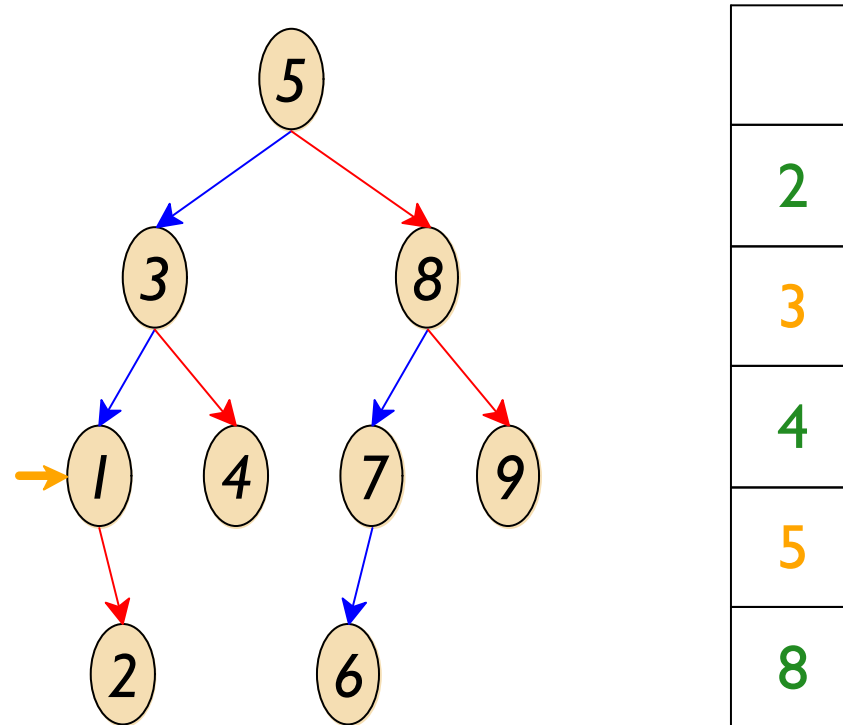
Depth-First Traversal (Inorder) via Stack



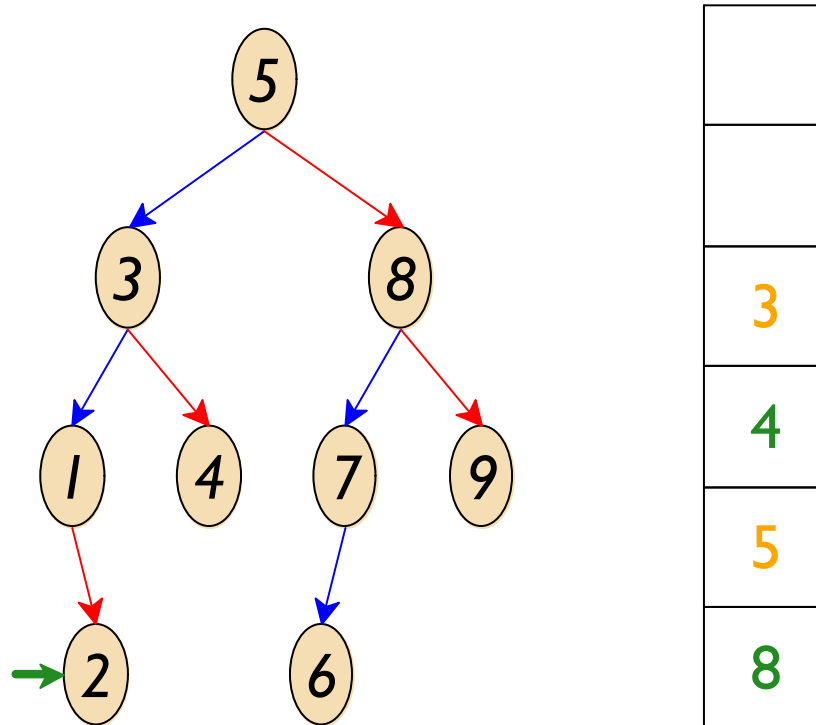
Depth-First Traversal (Inorder) via Stack



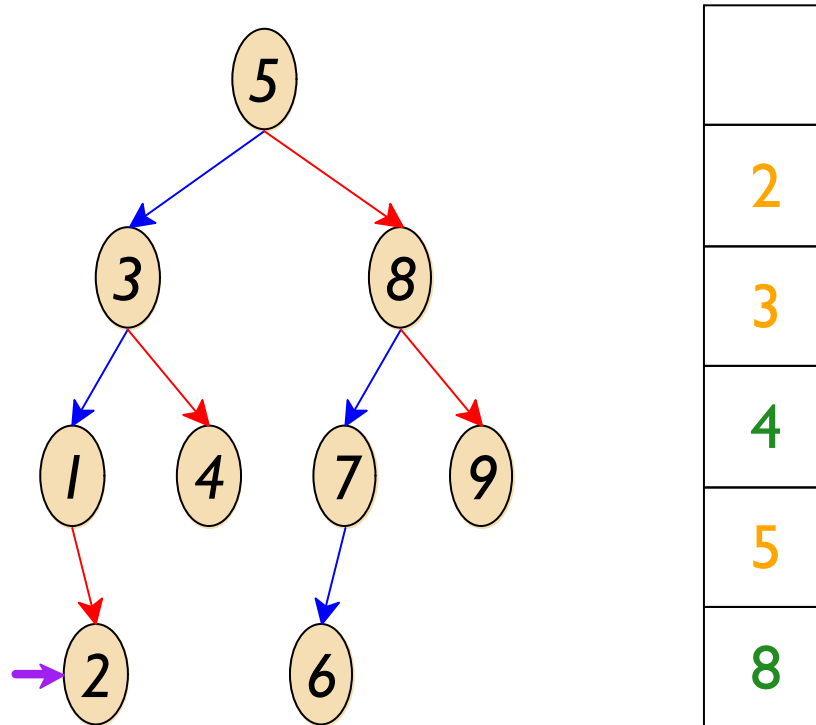
Depth-First Traversal (Inorder) via Stack



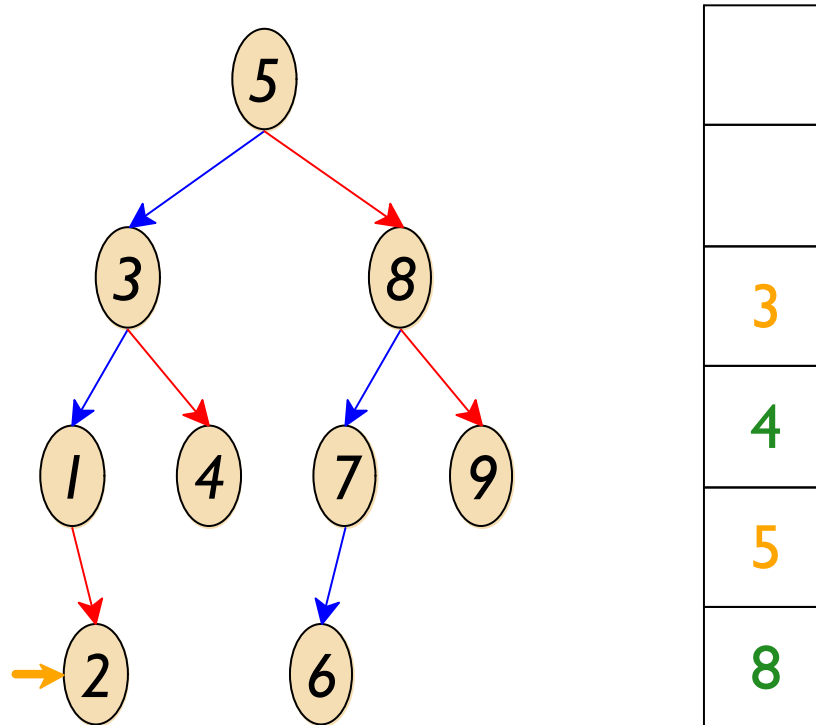
Depth-First Traversal (Inorder) via Stack



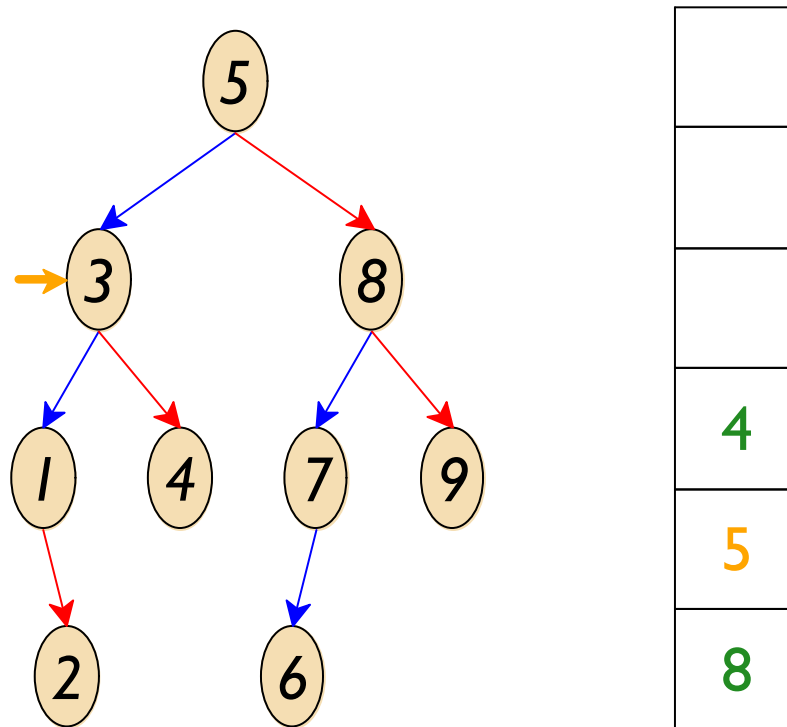
Depth-First Traversal (Inorder) via Stack



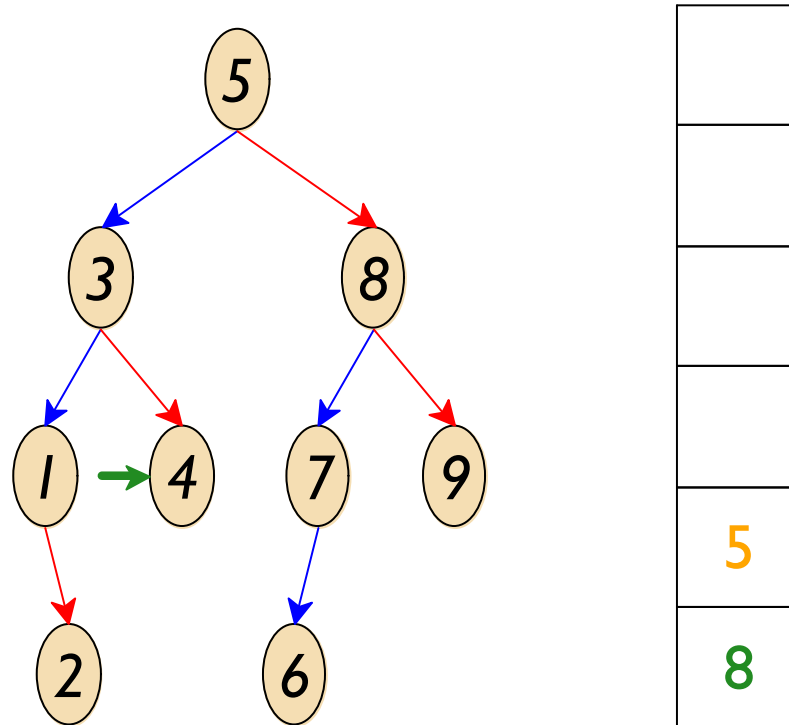
Depth-First Traversal (Inorder) via Stack



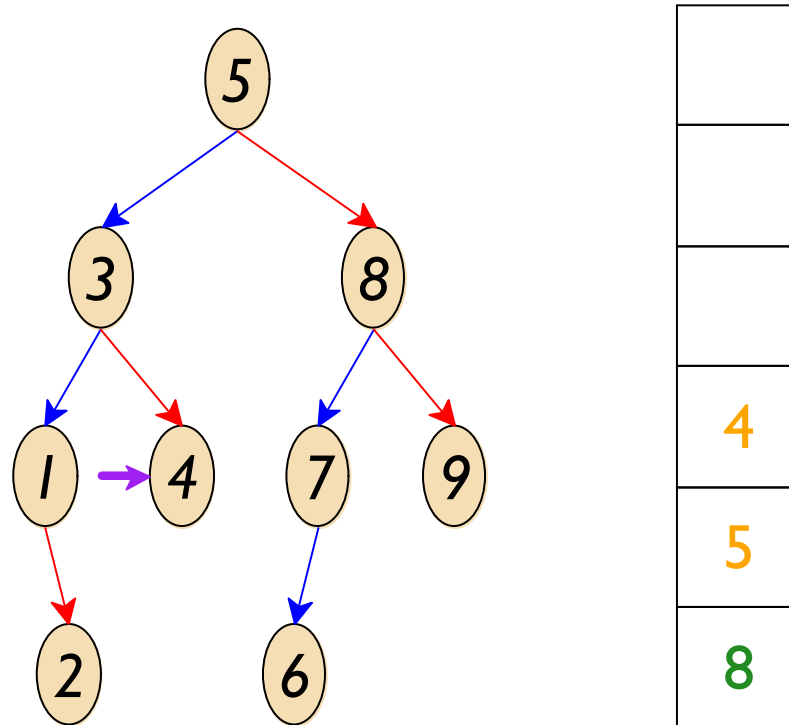
Depth-First Traversal (Inorder) via Stack



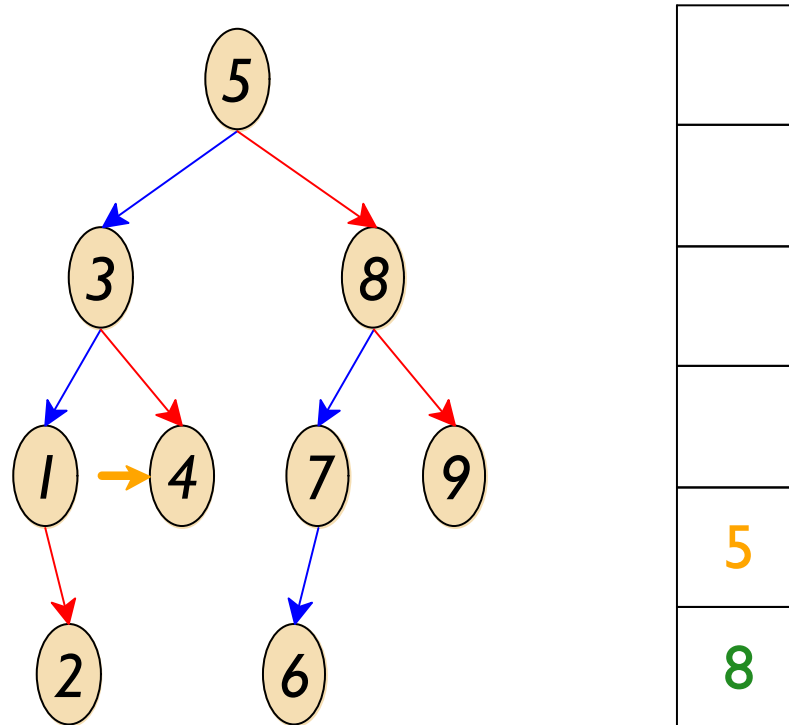
Depth-First Traversal (Inorder) via Stack



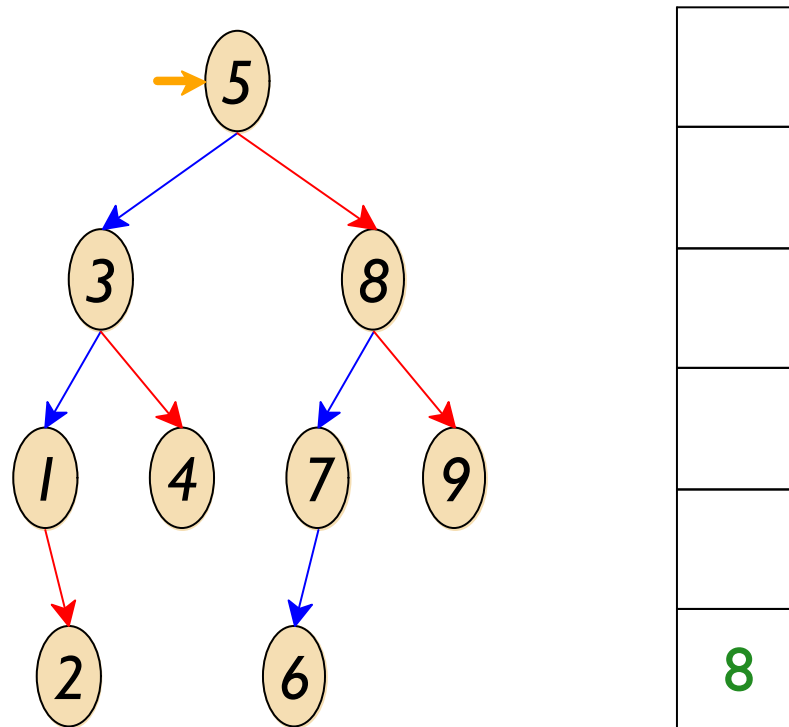
Depth-First Traversal (Inorder) via Stack



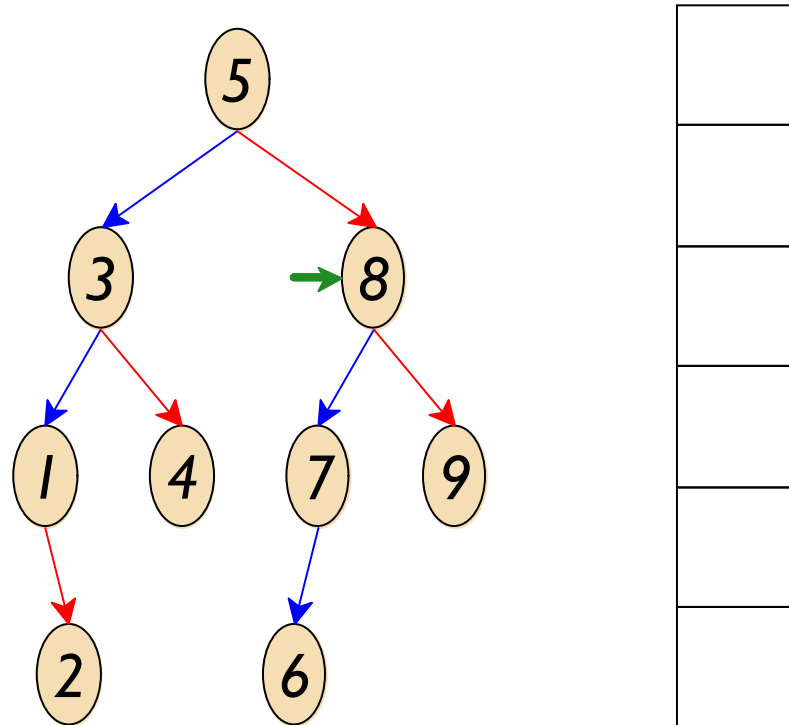
Depth-First Traversal (Inorder) via Stack



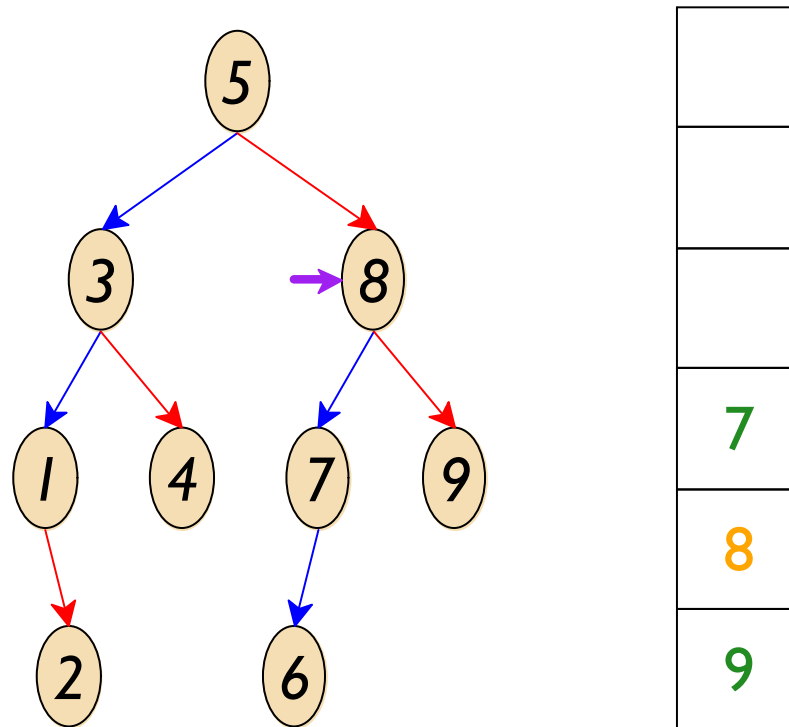
Depth-First Traversal (Inorder) via Stack



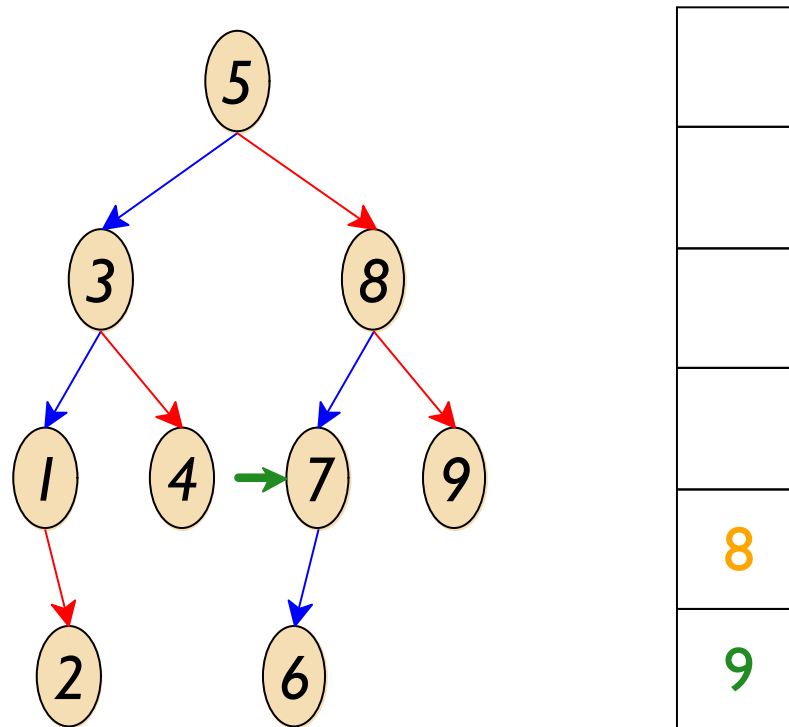
Depth-First Traversal (Inorder) via Stack



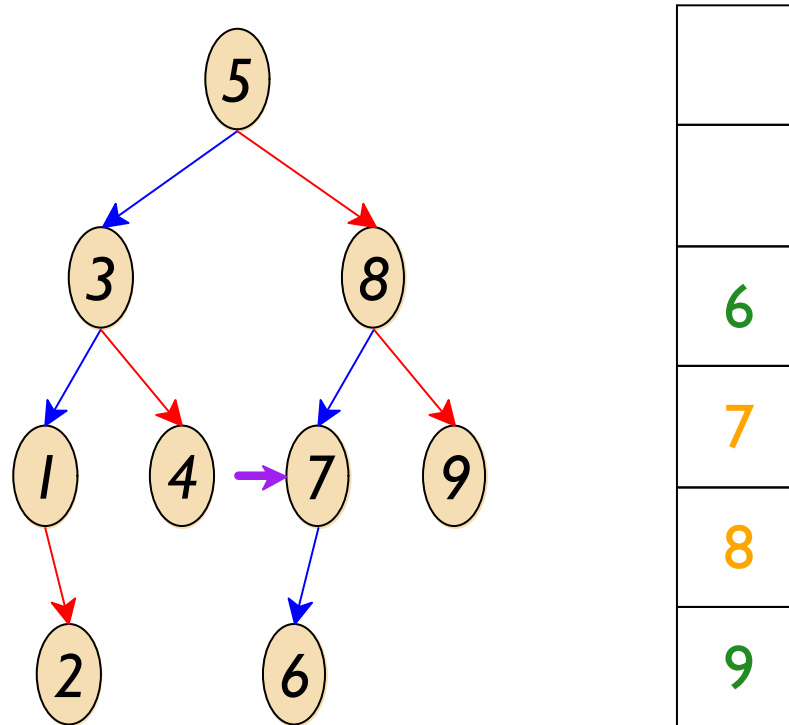
Depth-First Traversal (Inorder) via Stack



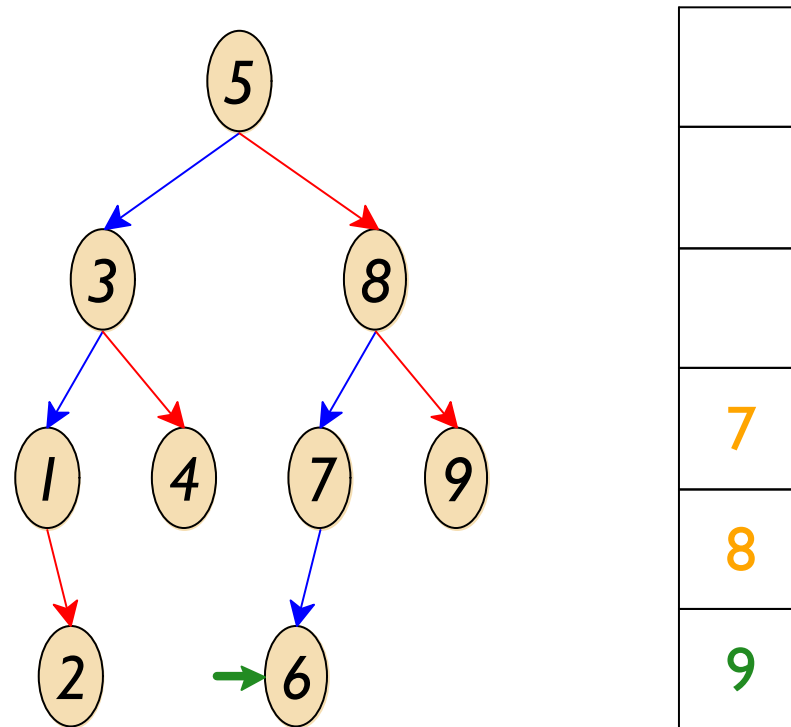
Depth-First Traversal (Inorder) via Stack



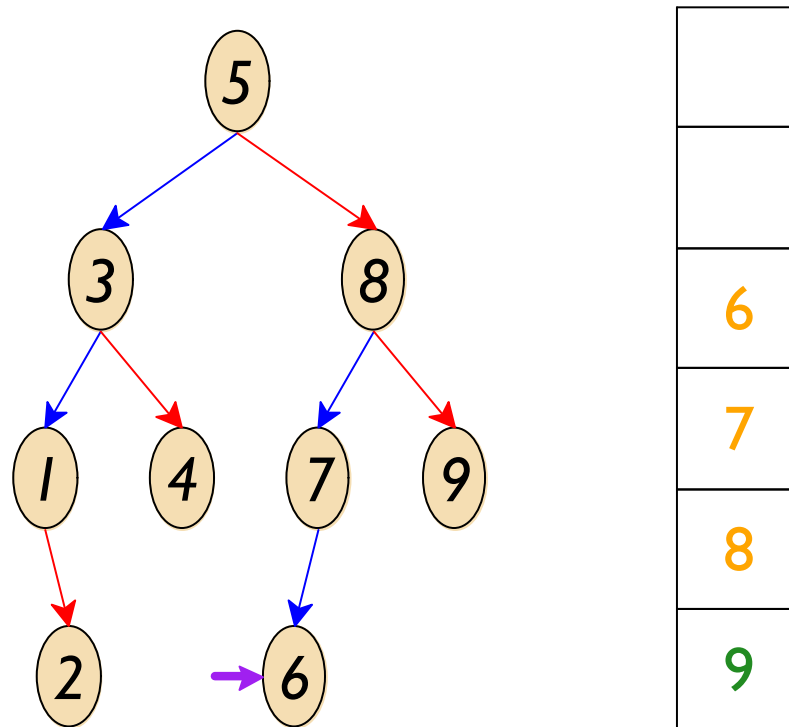
Depth-First Traversal (Inorder) via Stack



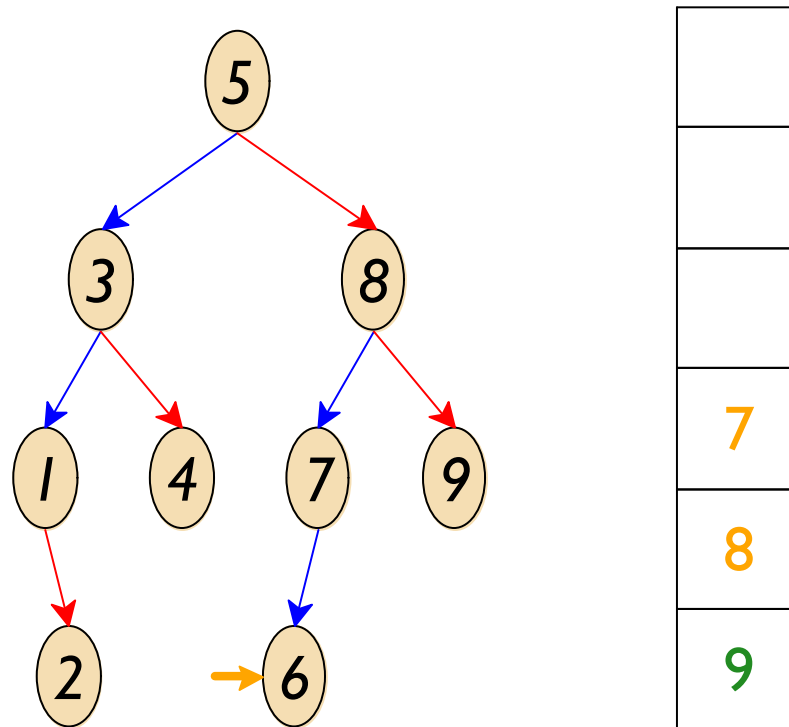
Depth-First Traversal (Inorder) via Stack



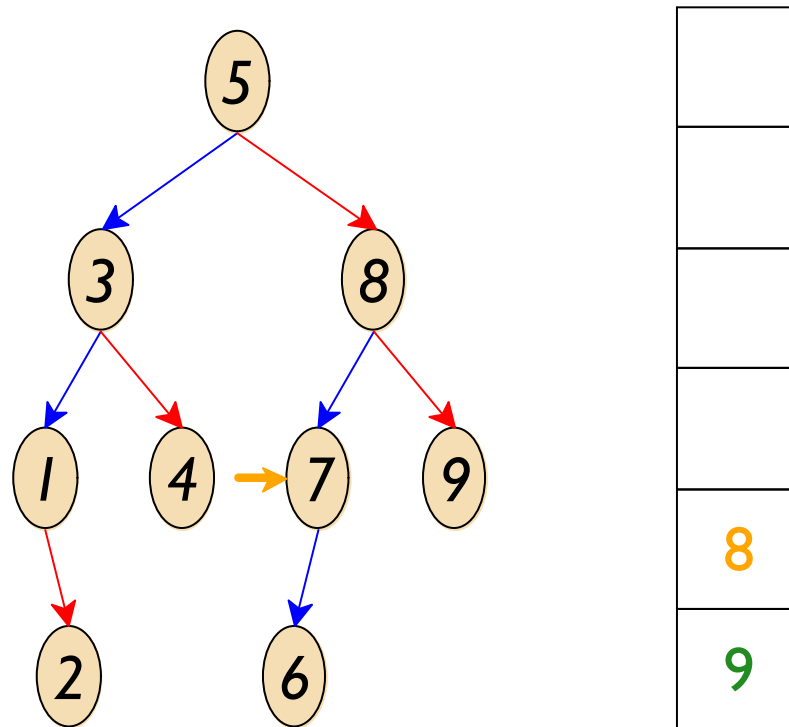
Depth-First Traversal (Inorder) via Stack



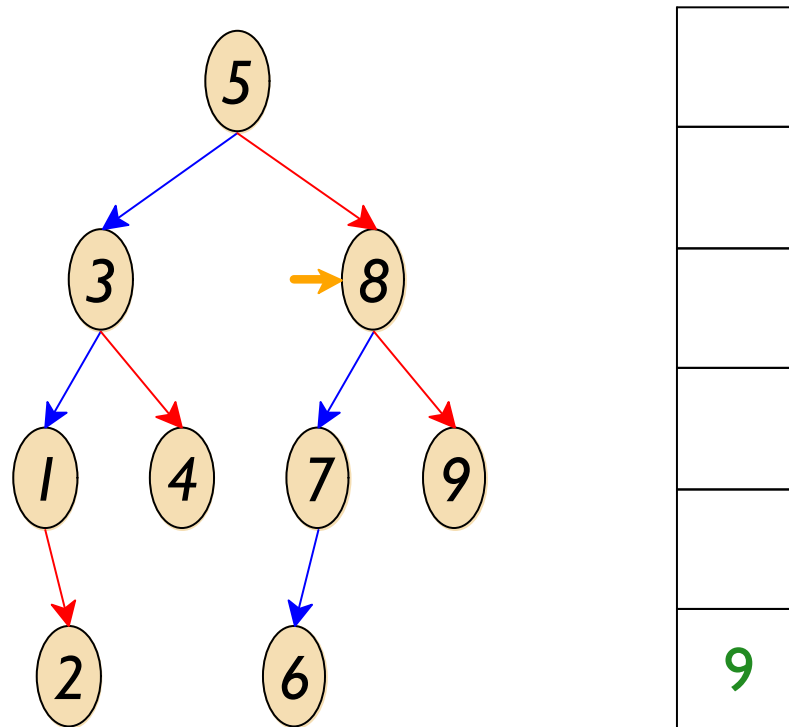
Depth-First Traversal (Inorder) via Stack



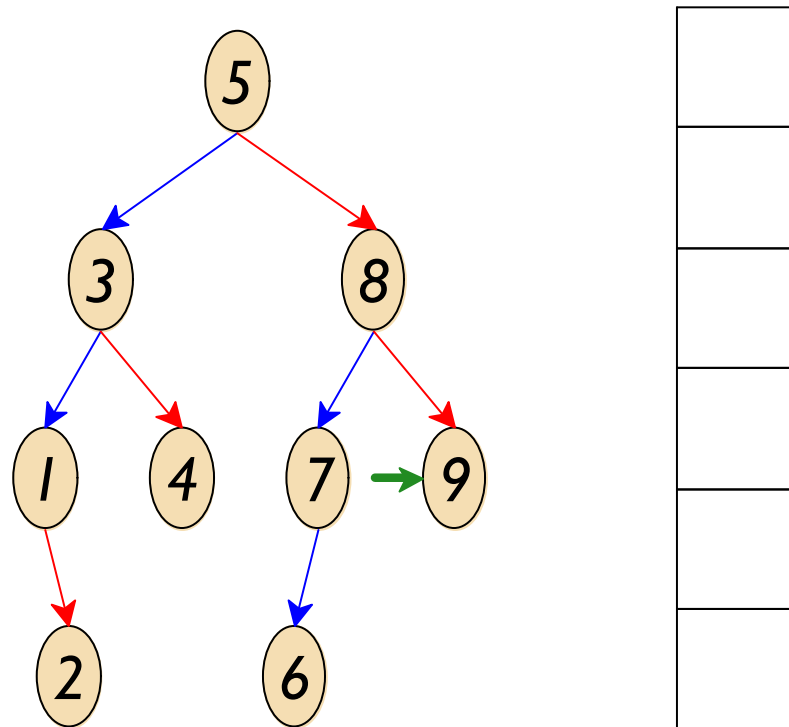
Depth-First Traversal (Inorder) via Stack



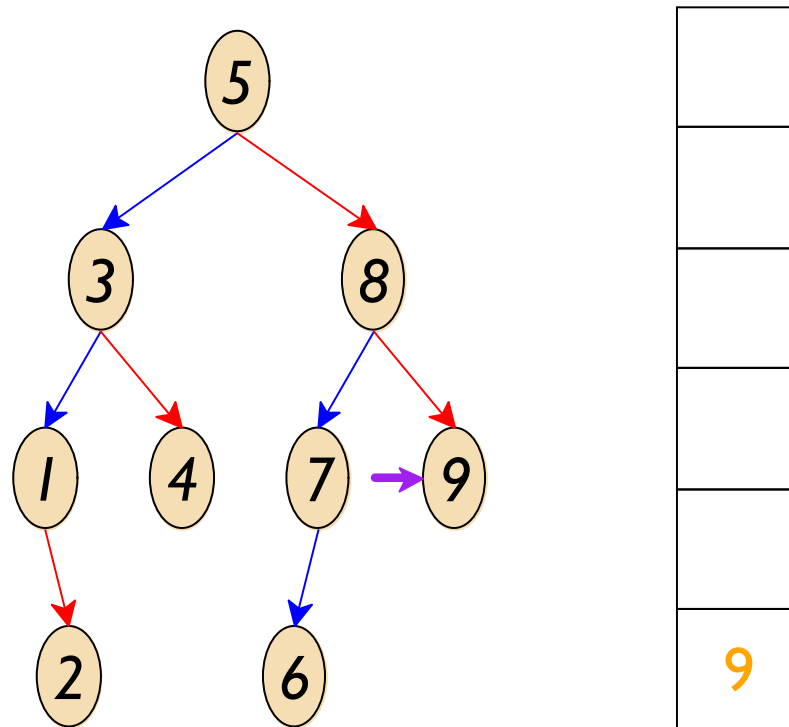
Depth-First Traversal (Inorder) via Stack



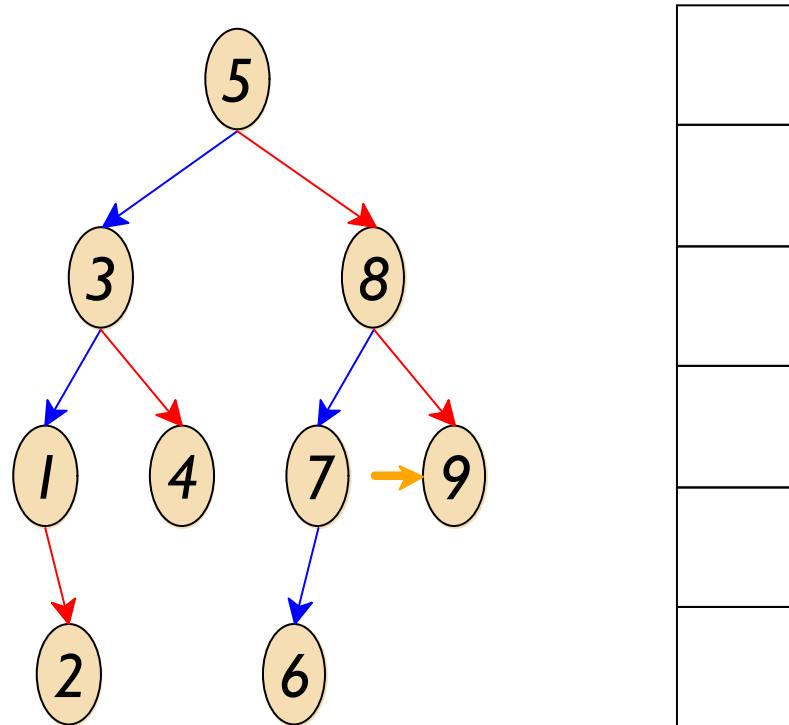
Depth-First Traversal (Inorder) via Stack



Depth-First Traversal (Inorder) via Stack



Depth-First Traversal (Inorder) via Stack



See `traverse.c`