

COMP 250

INTRODUCTION TO COMPUTER SCIENCE

Assignment Project Exam Help

<https://powcoder.com>

Week 11-2 : Tree Traversals

Giulia Alberini, Fall 2020

Slides adapted from Michael Langer's

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Tree traversals
- Depth first VS Breadth first
- Recursive and Non-recursive (with stack or with queue)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

TRAVERSALS
<https://powcoder.com>

Add WeChat powcoder

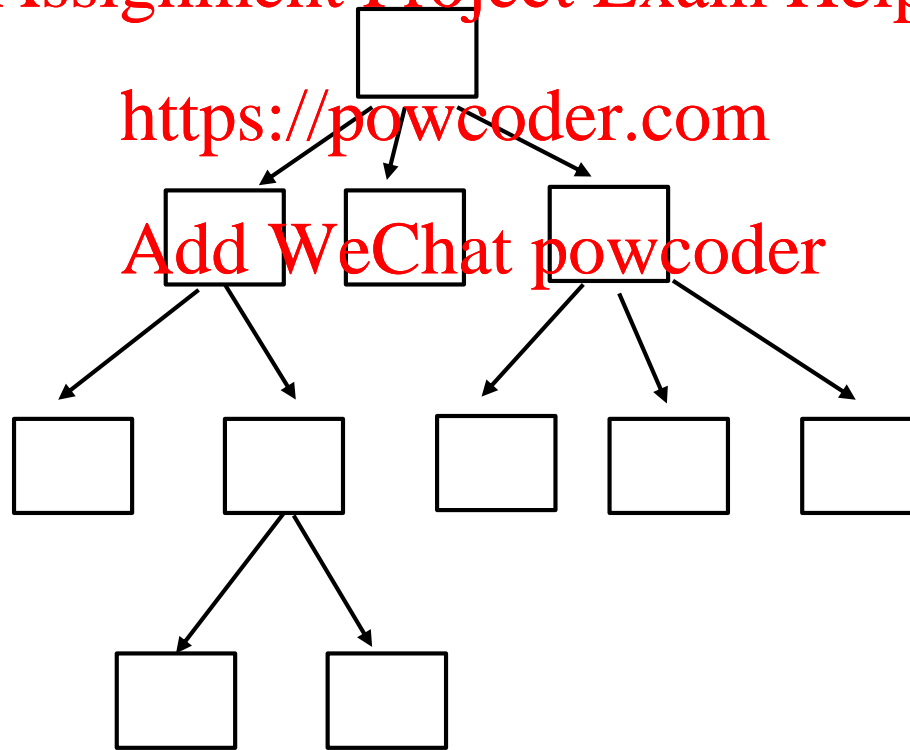
TREE TRAVERSAL

How to visit (enumerate, iterate through, traverse...) all the nodes of a tree ?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



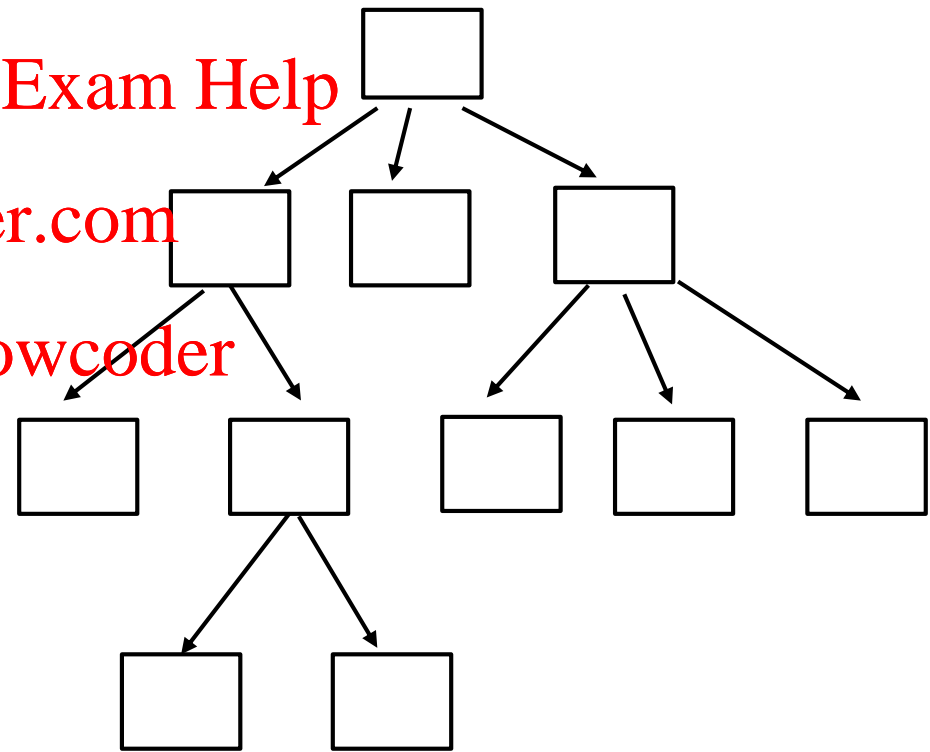
TREE TRAVERSAL – DEPTH FIRST “PREORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    visit root  
    for each child of root  
      depthfirst( child )  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



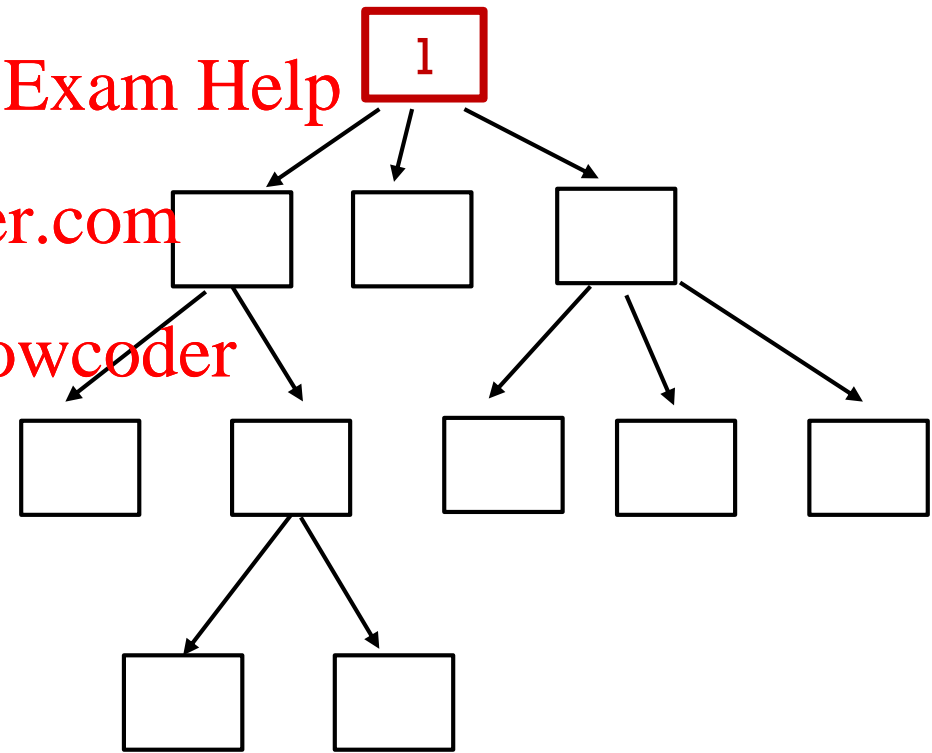
TREE TRAVERSAL – DEPTH FIRST “PREORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    visit root  
    for each child of root  
      depthfirst( child )  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



“preorder” traversal: visit the root before the children

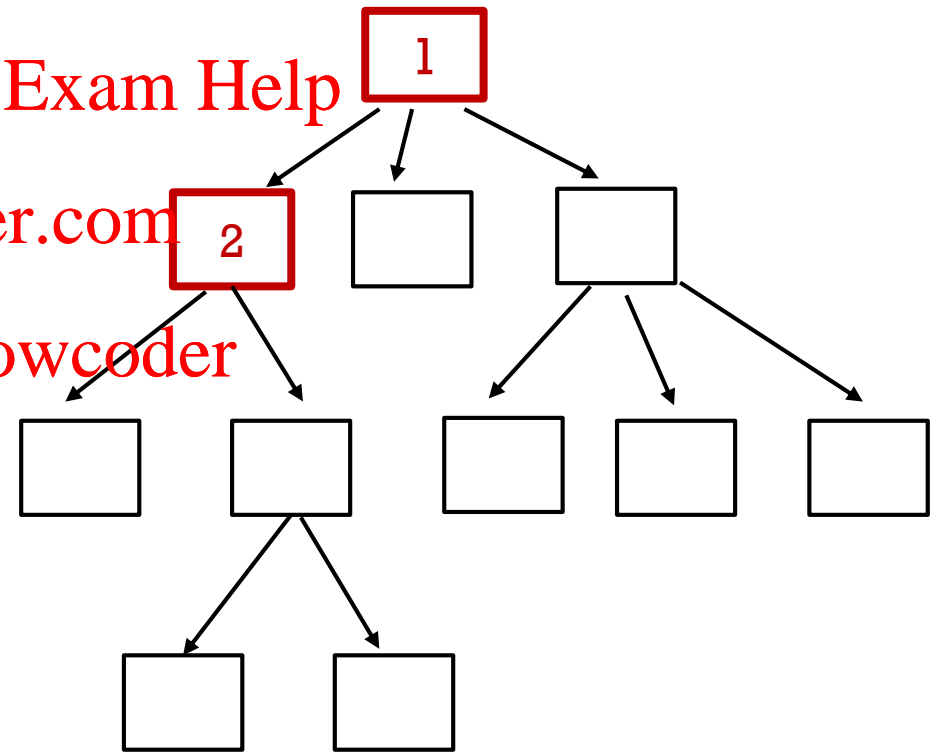
TREE TRAVERSAL – DEPTH FIRST “PREORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    visit root  
    for each child of root  
      depthfirst( child )  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Note that here we are assuming that we iterate through the children nodes from left to right.

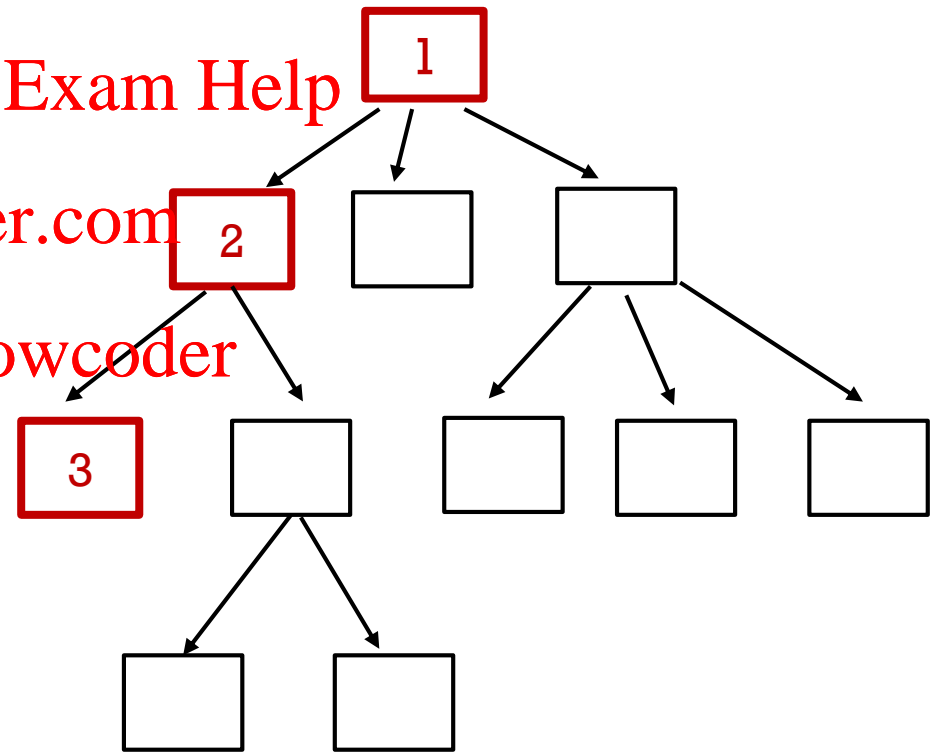
TREE TRAVERSAL – DEPTH FIRST “PREORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    visit root  
    for each child of root  
      depthfirst( child )  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



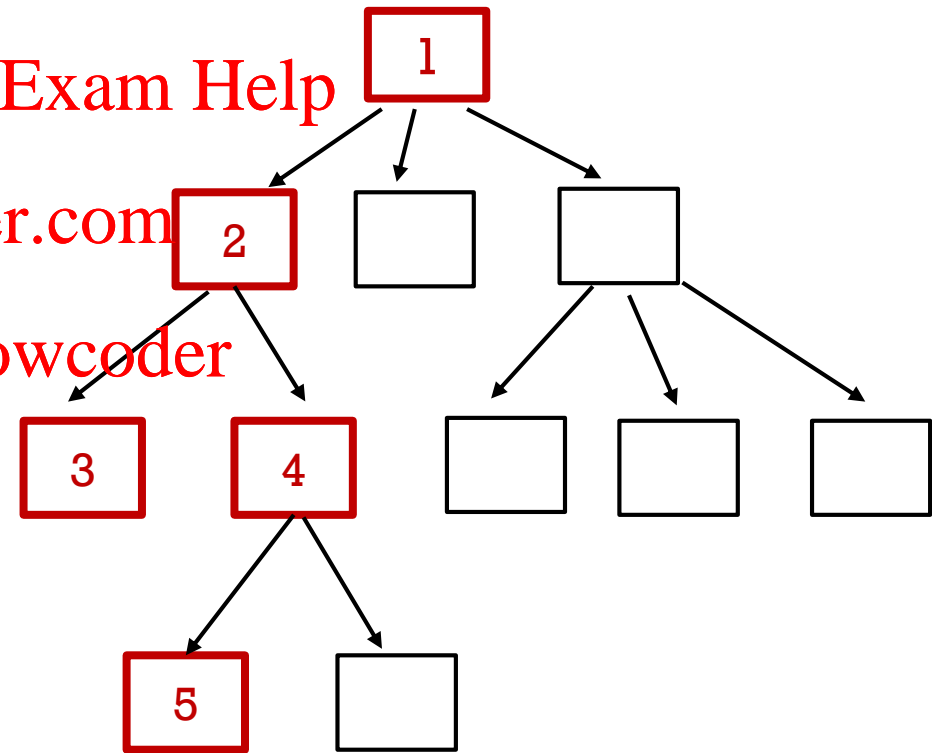
TREE TRAVERSAL – DEPTH FIRST “PREORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    visit root  
    for each child of root  
      depthfirst( child )  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



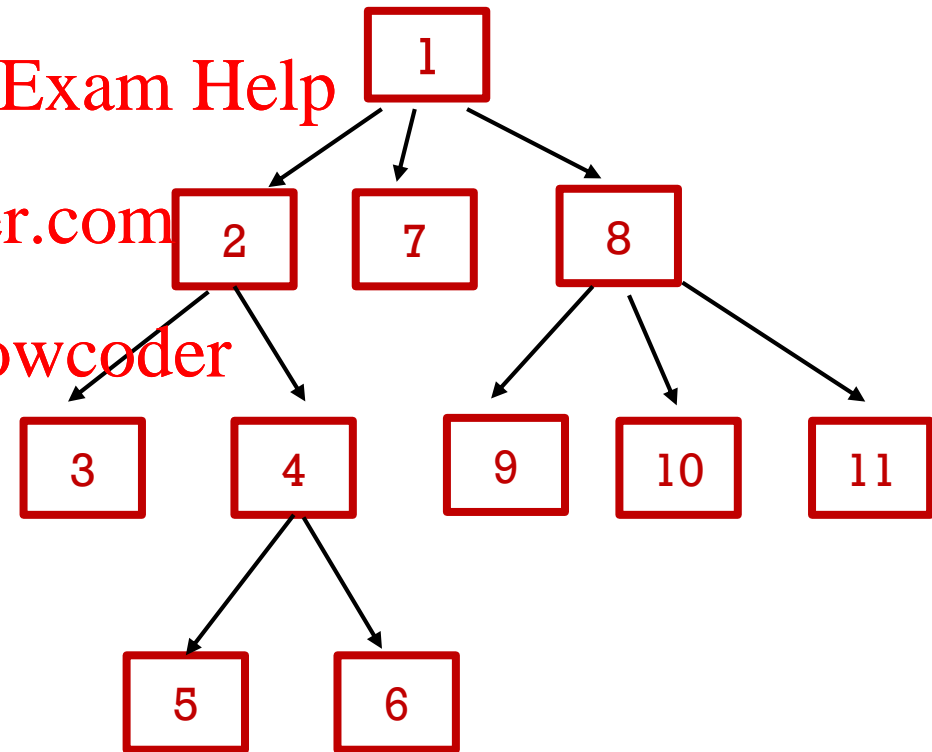
TREE TRAVERSAL – DEPTH FIRST “PREORDER”

```
depthFirst (root) {  
    if (root is not empty) {  
        visit root  
        for each child of root  
            depthfirst( child )  
    }  
}
```

Assignment Project Exam Help

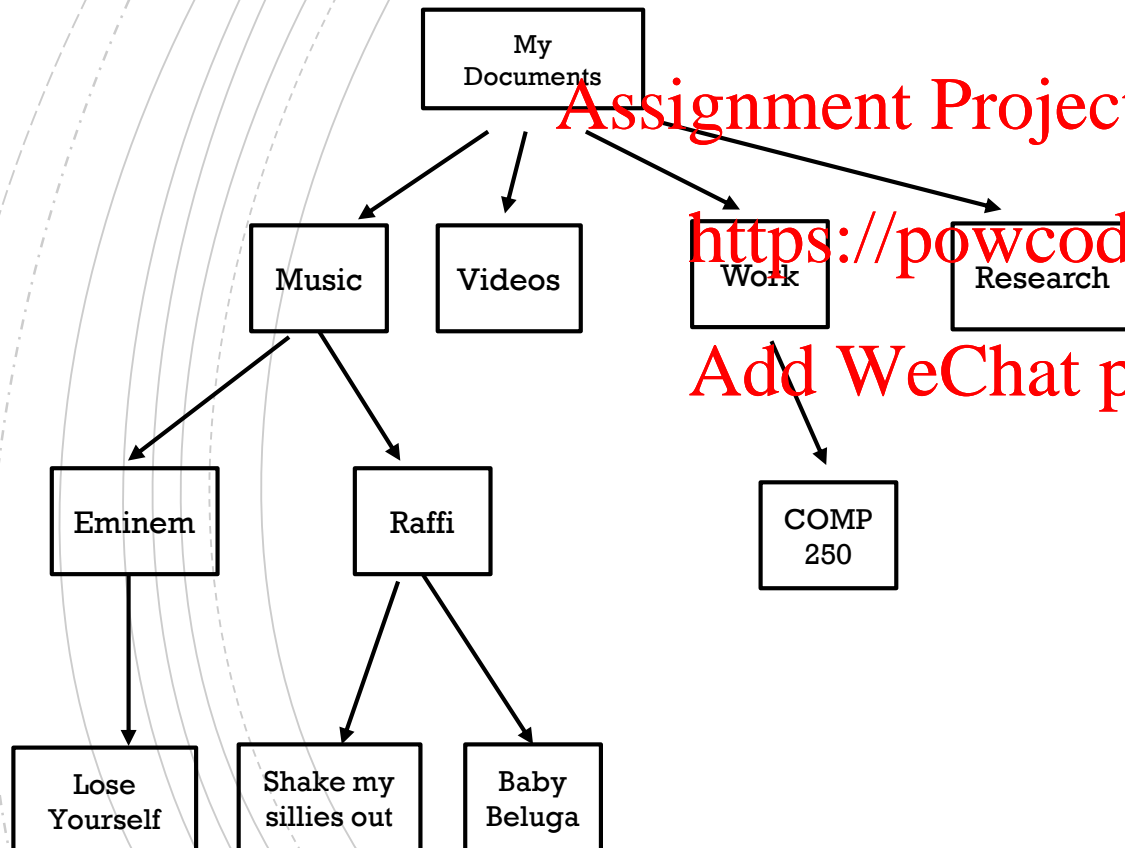
<https://powcoder.com>

Add WeChat powcoder



EXAMPLE OF USING A PREORDER TRAVERSAL

We would like to print a hierarchical file system
(visit = print directory or file name)



Documents (directory)
Music (directory)
Eminem (directory)
Lose Yourself (file)
Raffi (directory)
Shake My Sillies Out (file)
Baby Beluga (file)
Videos (directory)
:
Work (directory)
COMP250 (directory)
:
Research (directory)
:

“VISIT” A NODE

“Visit” implies that you do something at that node.

<https://powcoder.com>

Analogy: you aren't visiting London UK if you just fly through Heathrow.

TREE TRAVERSAL – DEPTH FIRST “POSTORDER”

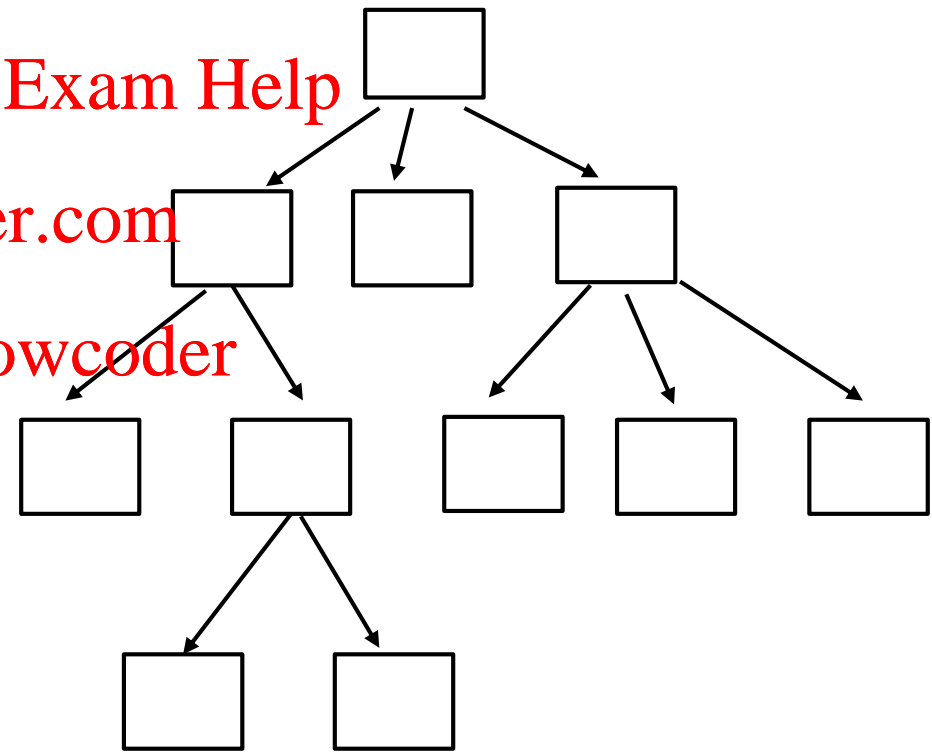
Q: Which node is visited first?

```
depthFirst (root) {  
  if (root is not empty) {  
    for each child of root {  
      depthfirst ( child )  
    }  
    visit root  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



“postorder” traversal: visit
the root after the children

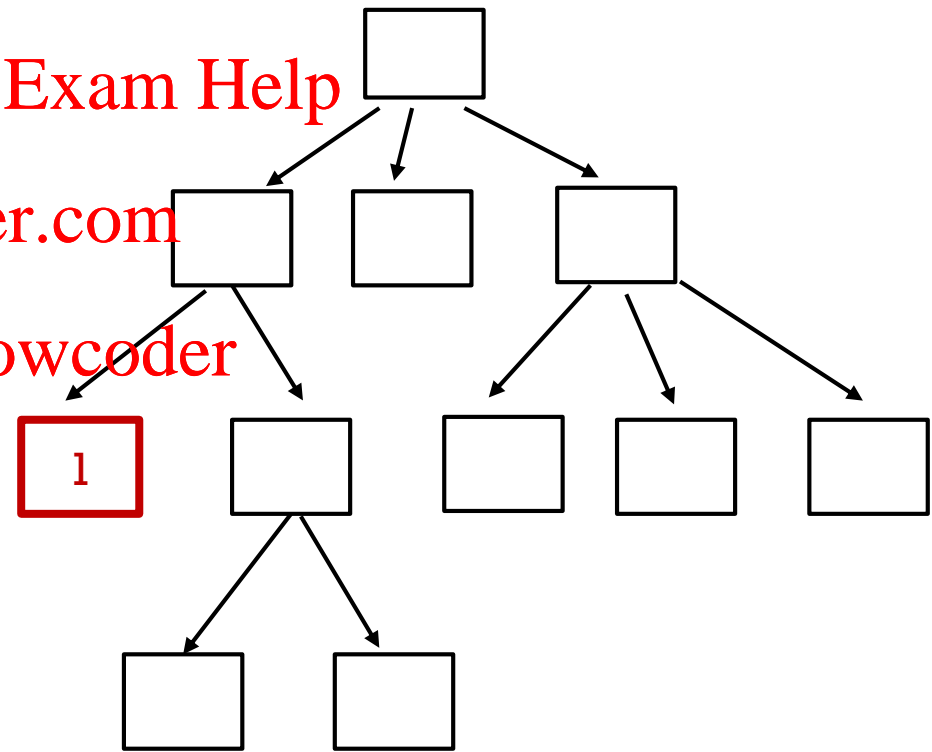
TREE TRAVERSAL – DEPTH FIRST “POSTORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    for each child of root  
      depthfirst ( child )  
    visit root  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



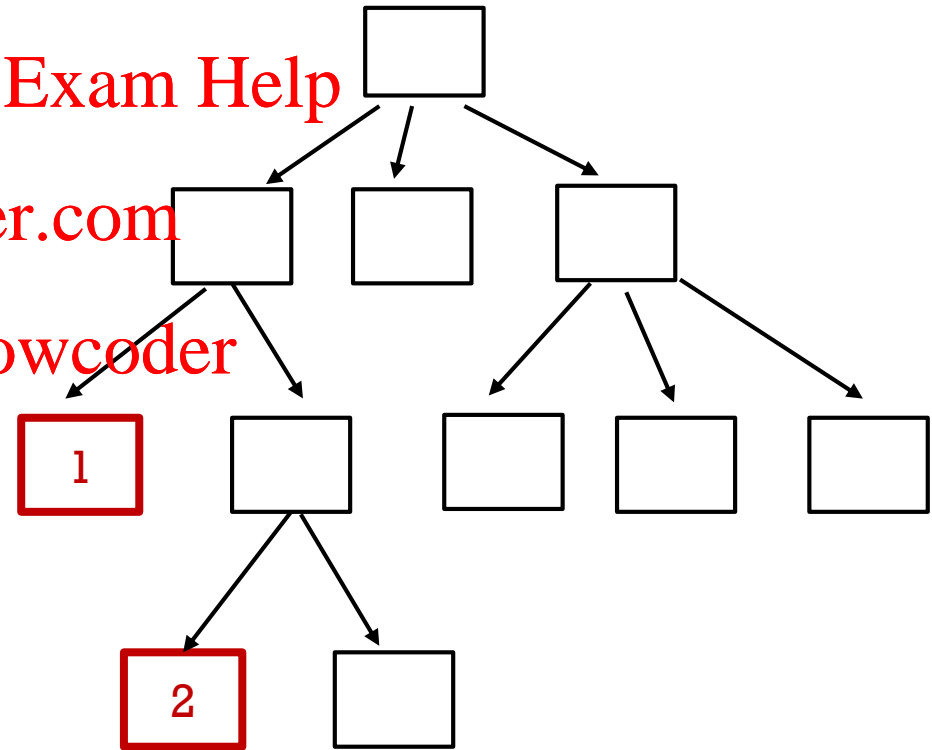
TREE TRAVERSAL – DEPTH FIRST “POSTORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    for each child of root  
      depthfirst ( child )  
    visit root  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



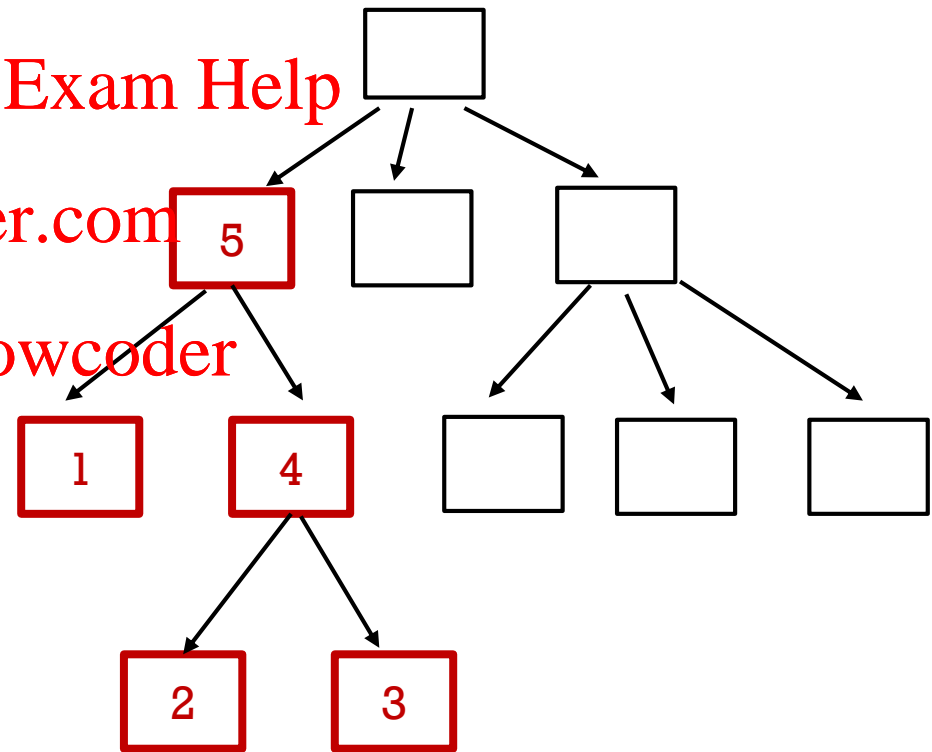
TREE TRAVERSAL – DEPTH FIRST “POSTORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    for each child of root {  
      depthfirst ( child )  
    }  
    visit root  
  }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



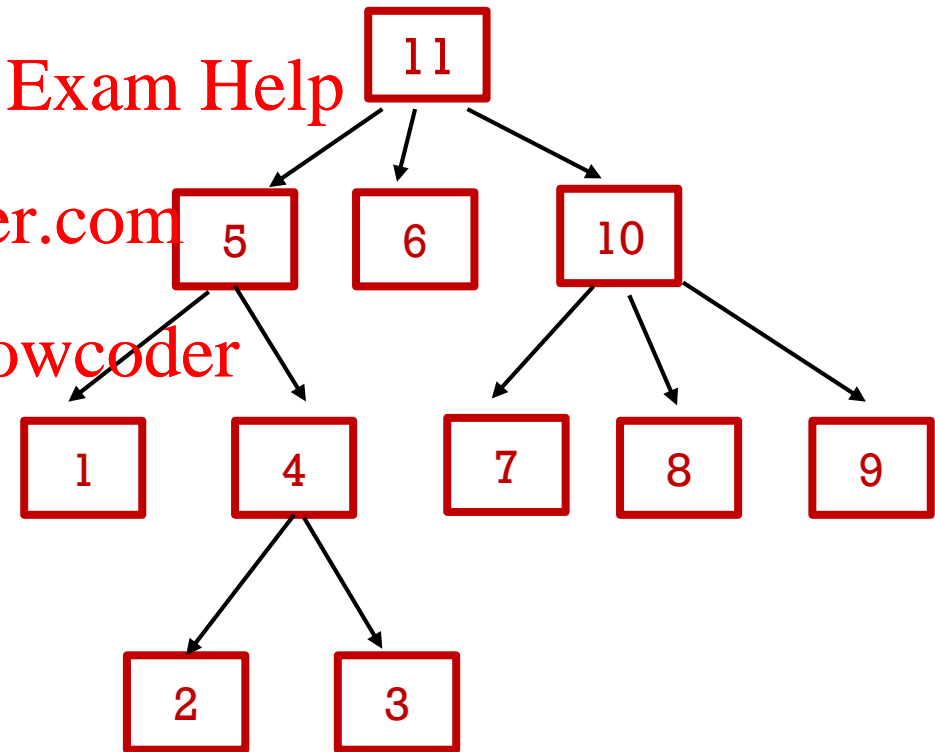
TREE TRAVERSAL – DEPTH FIRST “POSTORDER”

```
depthFirst (root) {  
  if (root is not empty) {  
    for each child of root {  
      depthfirst ( child )  
    }  
    visit root  
  }  
}
```

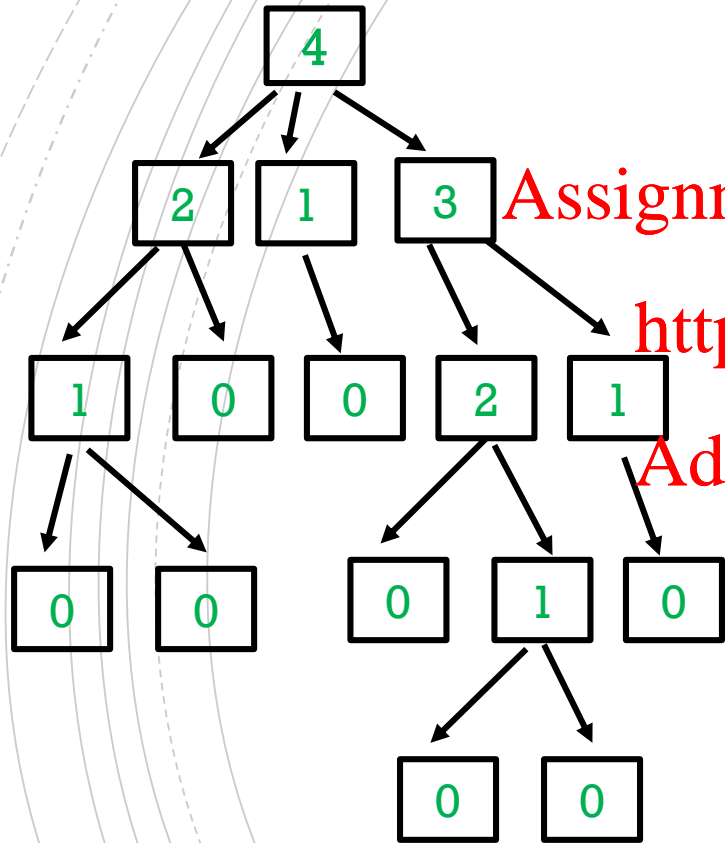
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



EXAMPLE 1 OF USING A POSTORDER TRAVERSAL: height(v)

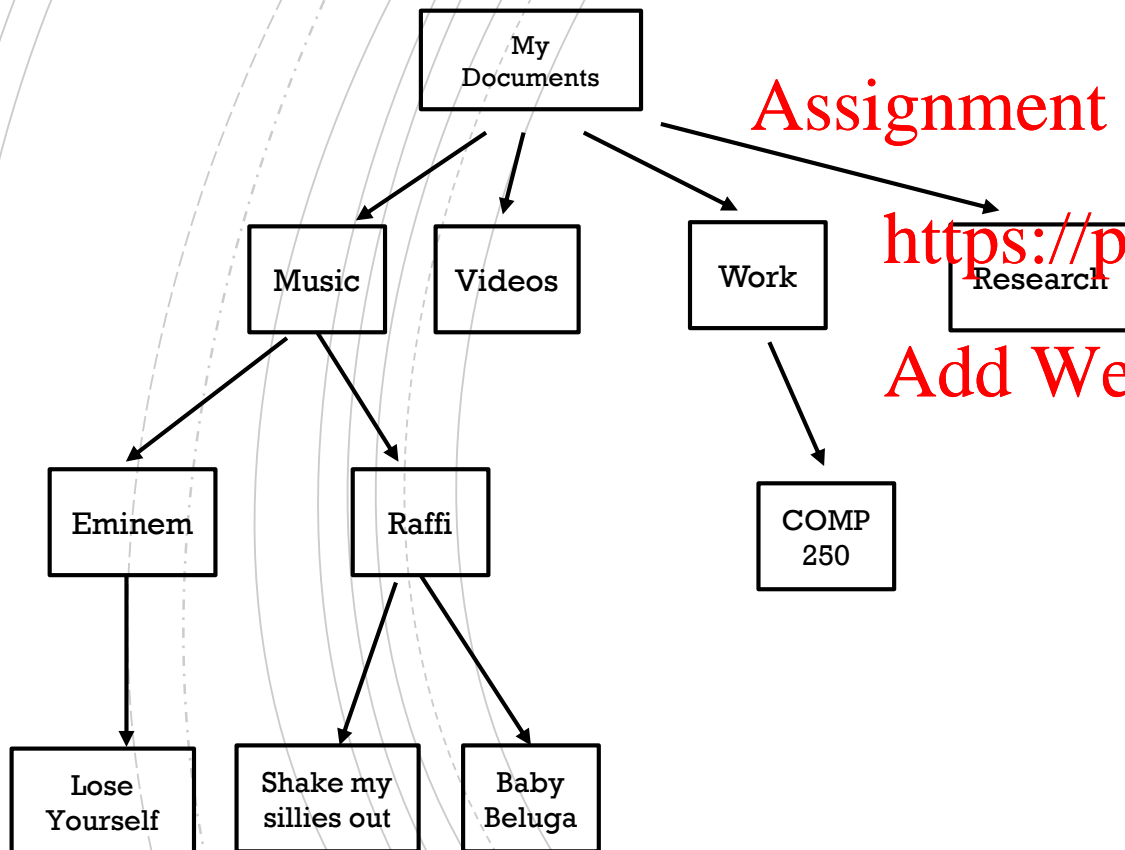


```
height(v) {  
    if (v is a leaf)  
        return 0  
    else  
        h = 0  
        for each child w of v  
            h = max(h, height(w))  
        return 1 + h  
    }  
}
```

visit = return value of height

EXAMPLE 2 OF USING A POSTORDER TRAVERSAL

What is the total number of bytes in all files in a directory?



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
numBytes(v) {  
    if (v is a leaf)  
        return number of bytes at v  
    else {  
        sum = 0  
        for each child w of v  
            sum += numBytes(w)  
        return sum  
    }  
}
```

visit = determining the number of bytes for a node, e.g. If we were to store 'sum' at the node.

depthFirst () – PREORDER VS POSTORDER TRAVERSAL

```
depthFirst (root) {  
    if (root is not empty) {  
        visit root  
        for each child of root  
            depthfirst( child )  
    }  
}
```

```
depthFirst (root) {  
    if (root is not empty) {  
        for each child of root  
            depthfirst( child )  
        visit root  
    }  
}
```

Assignment Project Exam Help

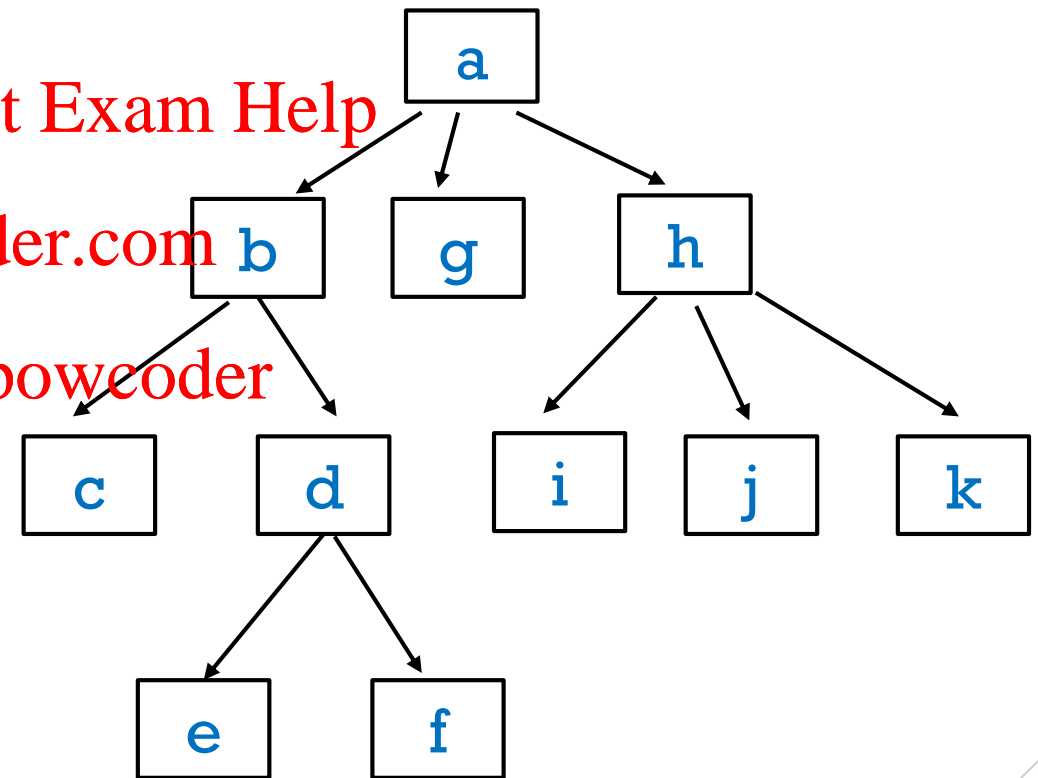
<https://powcoder.com>

Add WeChat powcoder

CALL SEQUENCE OF `depthFirst()`

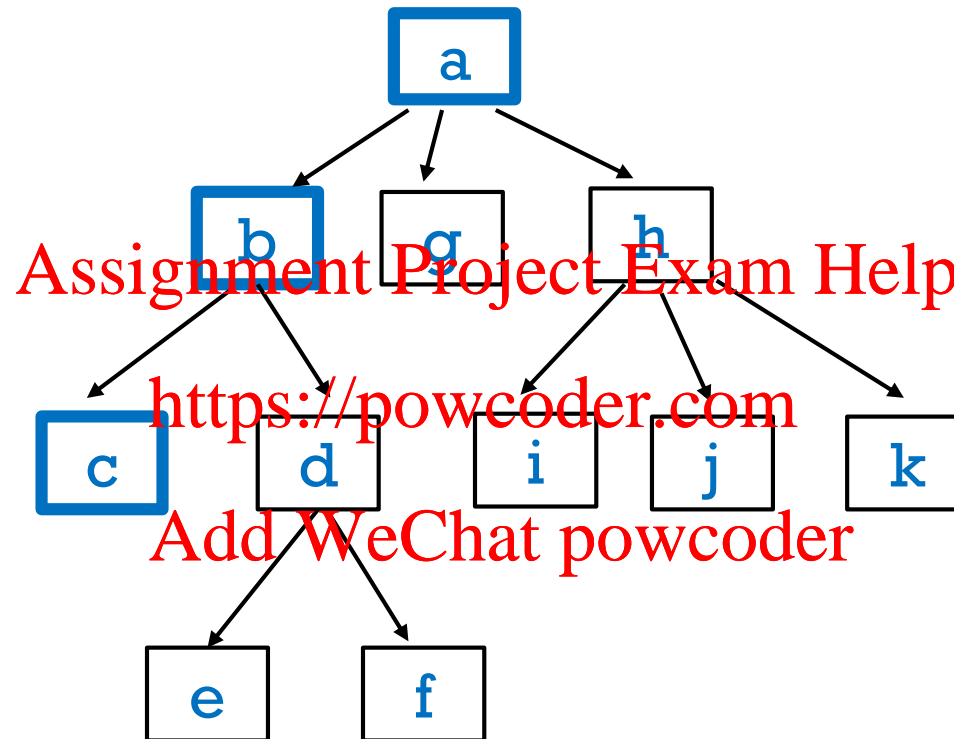
When we call `depthFirst(root)`, the same call sequence occurs for preorder vs postorder implementation.

In the example on the right, the letter order corresponds to `depthFirst(root)` call order.



Note that the call stack stores information about the active method calls in a program.

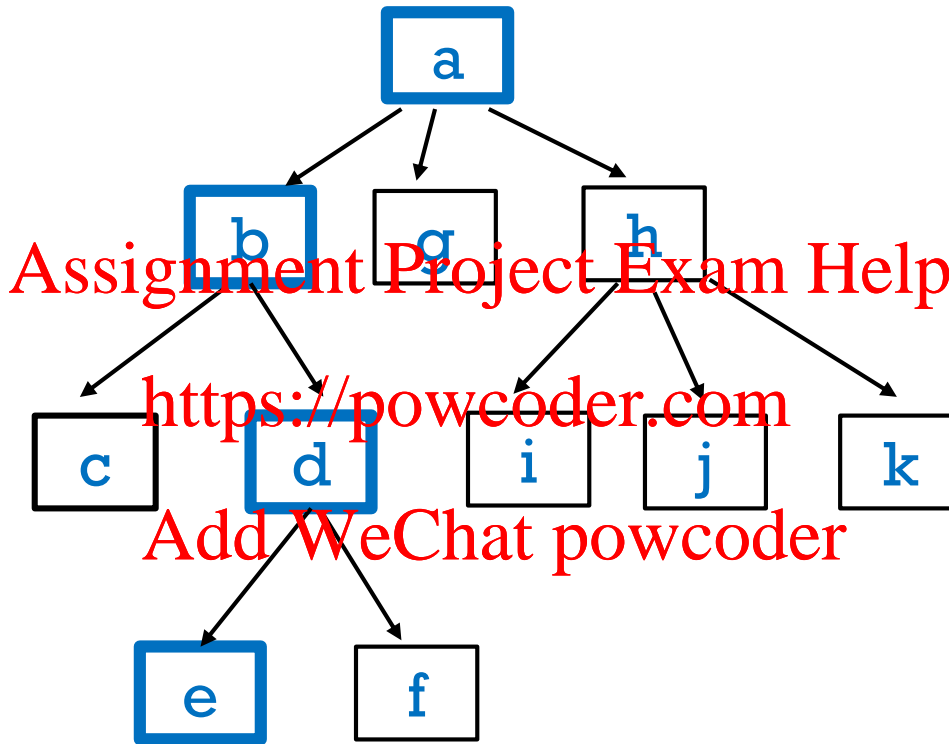
CALL STACK FOR depthFirst(root)



a a a
b b
c

Note that the call stack stores information about the active method calls in a program.

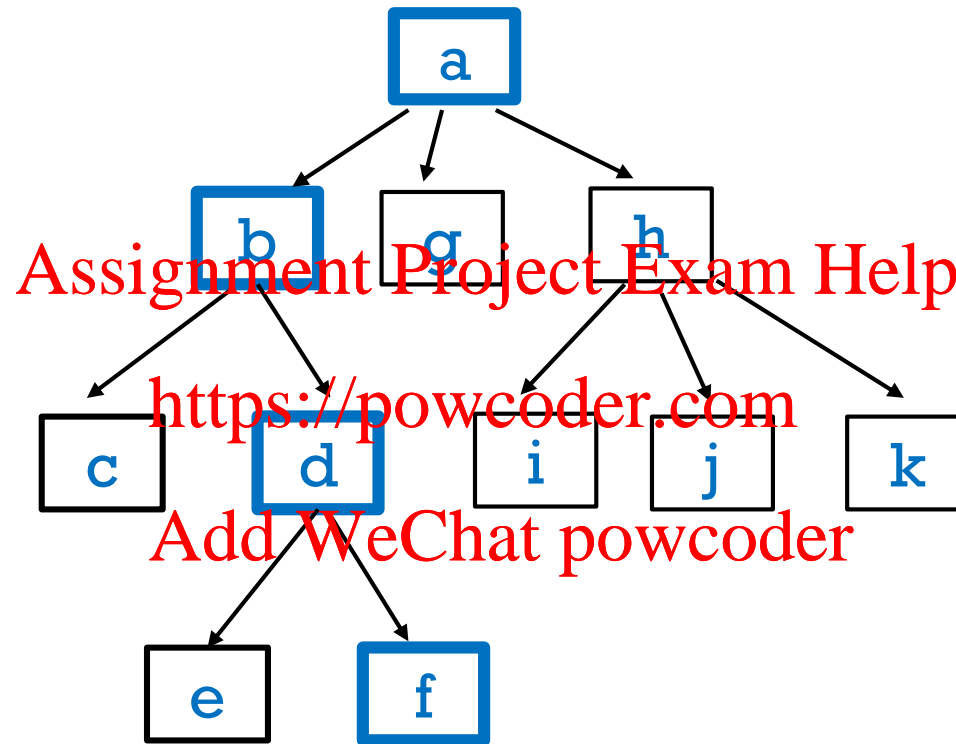
CALL STACK FOR depthFirst(root)



a a b b c b d b e d
a a a a a a

Note that the call stack stores information about the active method calls in a program.

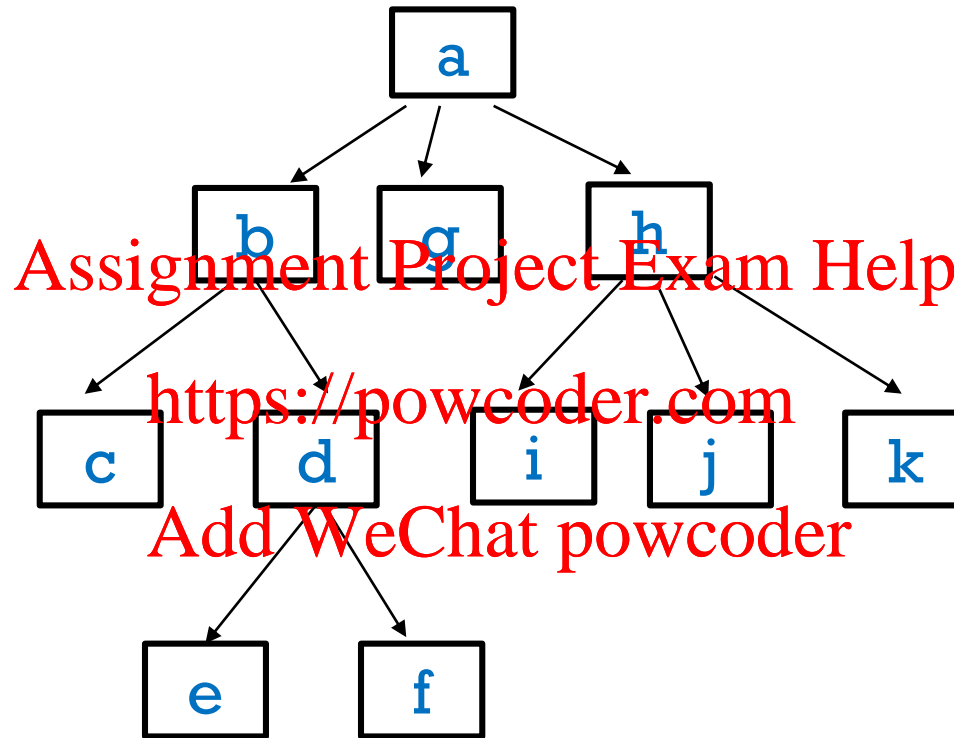
CALL STACK FOR depthFirst(root)



				e	f	
		c	d	d	d	
	b	b	b	b	b	
a	a	a	a	a	a	a

Note that the call stack stores information about the active method calls in a program.

CALL STACK FOR depthFirst(root)

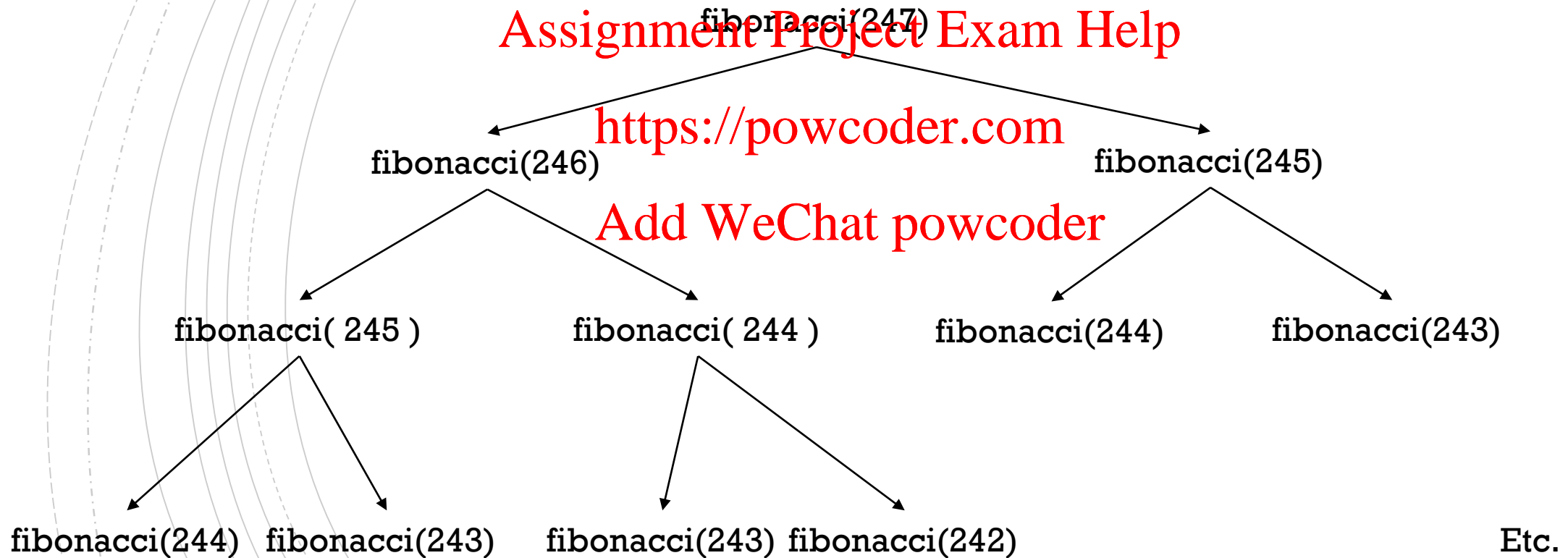


Notation: the letters indicate the call order of `depthFirst(root)`

a a b a c a b b d a d d e a d d f a b g a h a h a h j a h k a a

EXAMPLE

We used a tree to represent the call stack of the recursive Fibonacci method.



TREE TRAVERSAL IMPLEMENTATIONS

Recursive

- depth first (pre- versus post-order)

Assignment Project Exam Help

<https://powcoder.com>

Non-Recursive

- using a stack
- using a queue

Add WeChat powcoder

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
  
    while (s.empty() == false) {  
        node = s.pop()  
        visit node  
        if (node.left != null) s.push(node.left)  
        if (node.right != null) s.push(node.right)  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```

Assignment Project Exam Help

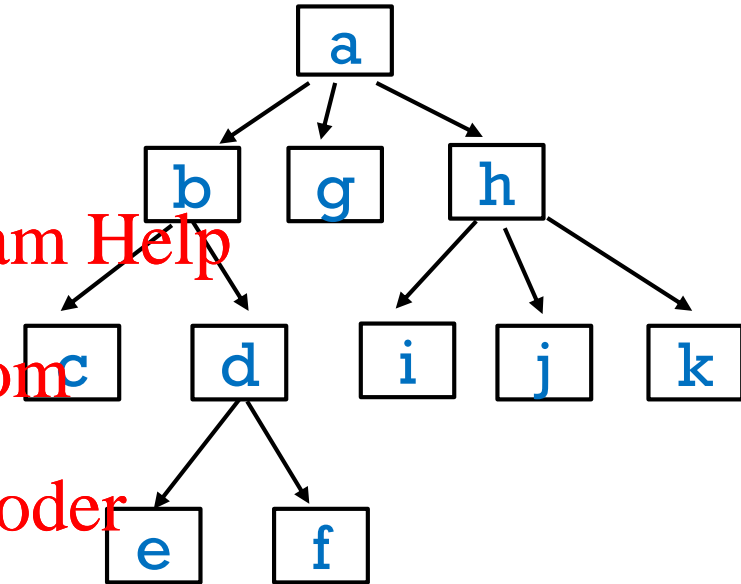
<https://powcoder.com>

Add WeChat powcoder

What is the order in which
the nodes are visited?

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



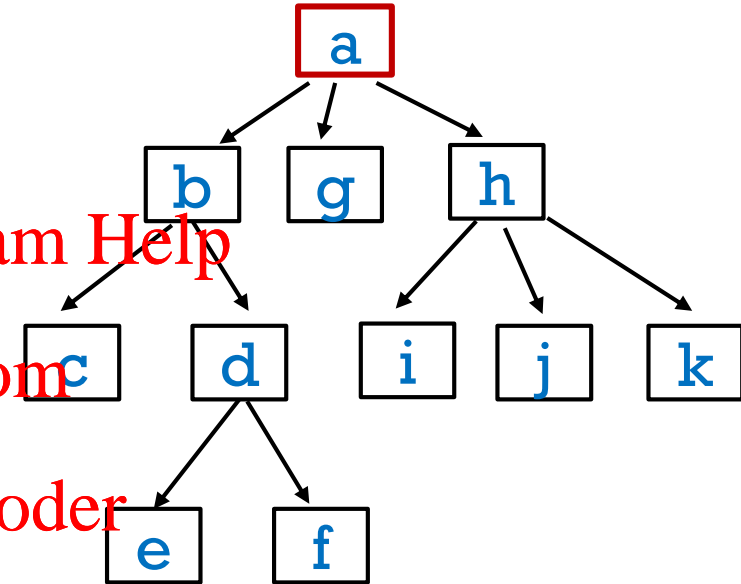
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

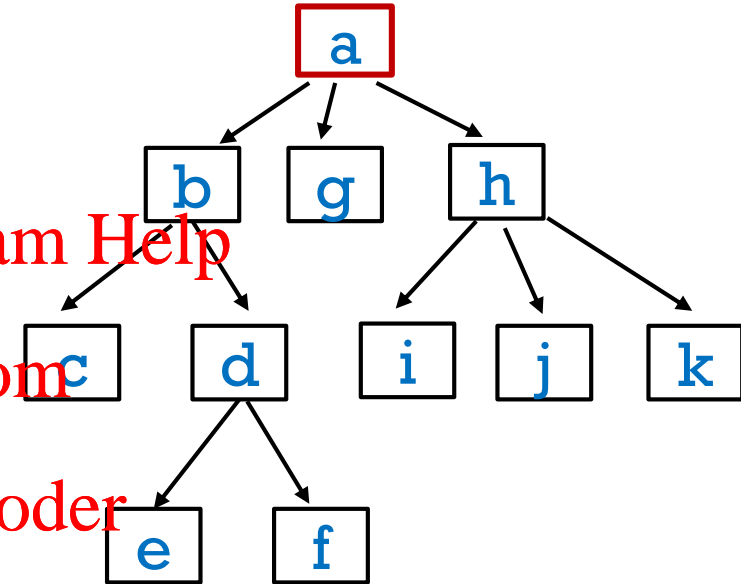
TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



TREE TRAVERSAL – WITH A STACK

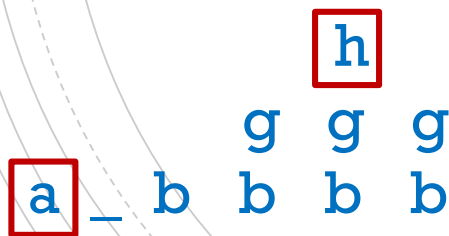
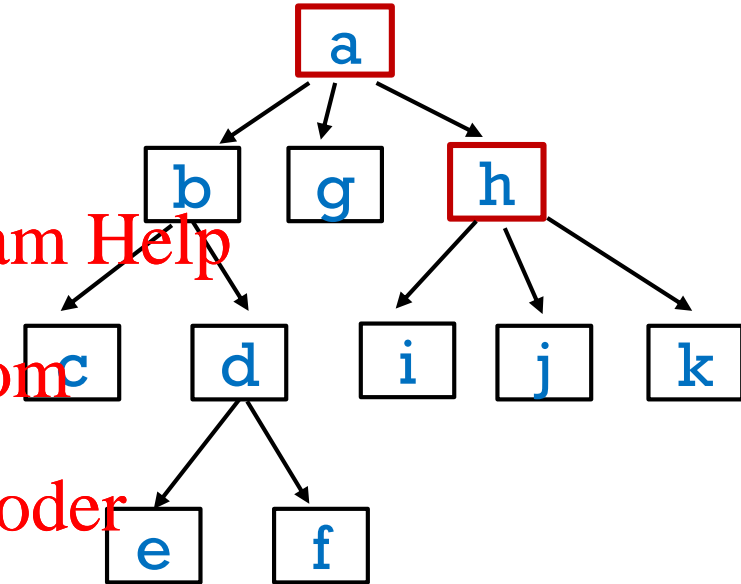
```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



a _ b b b h g g

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



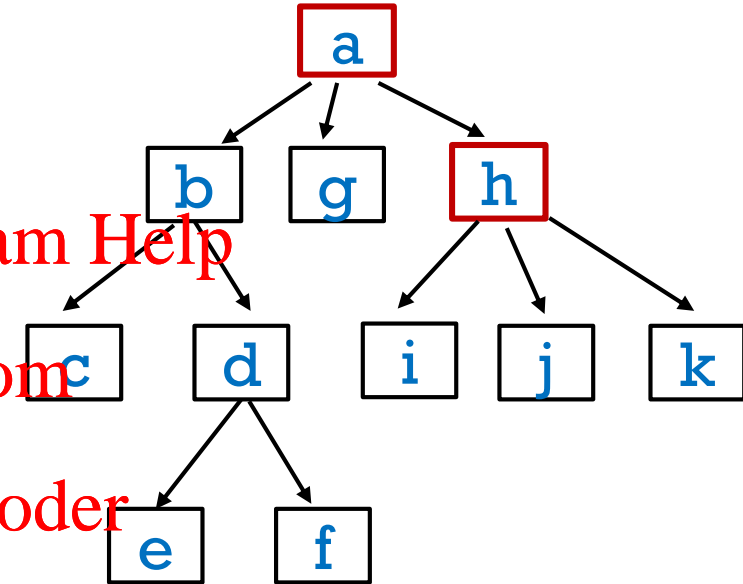
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



Sequence of nodes visited (root to leaf):

a	—	b	b	b	b	b	b	k
		g	g	g	g	g	g	j
			h		i	i	i	j

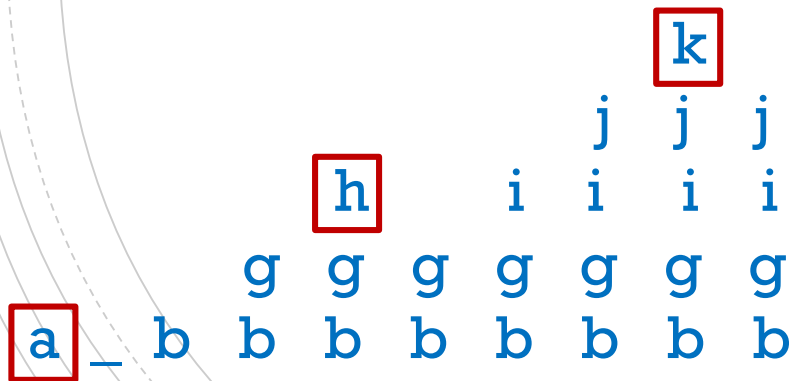
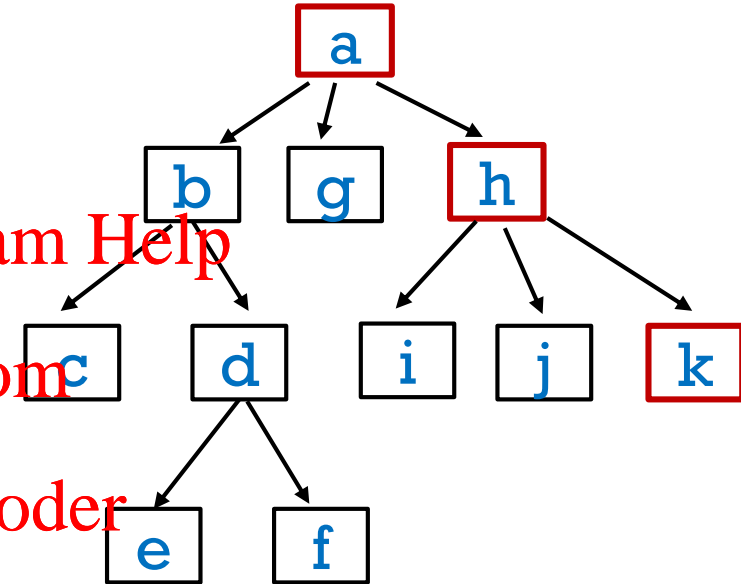
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

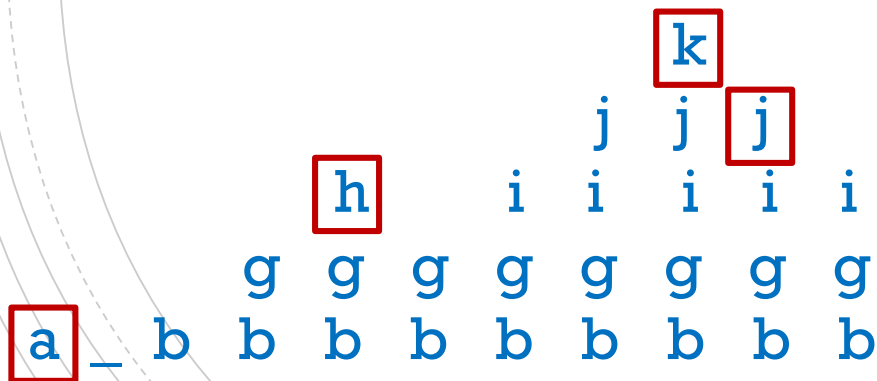
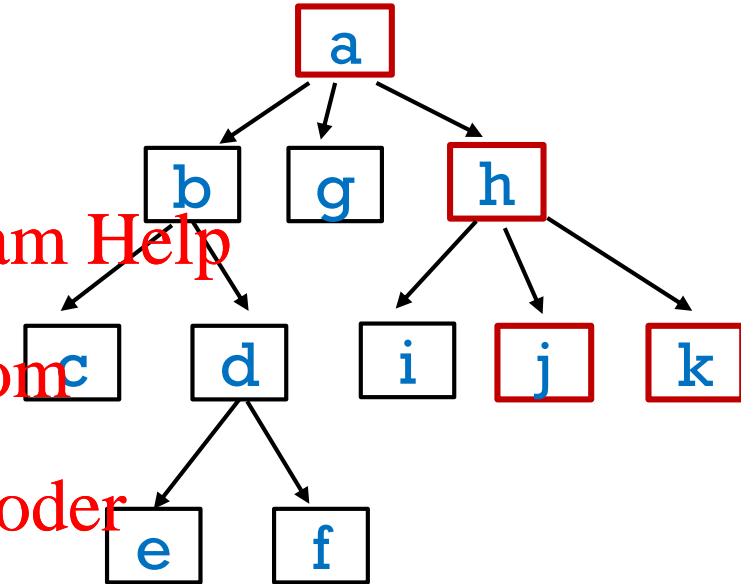
TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



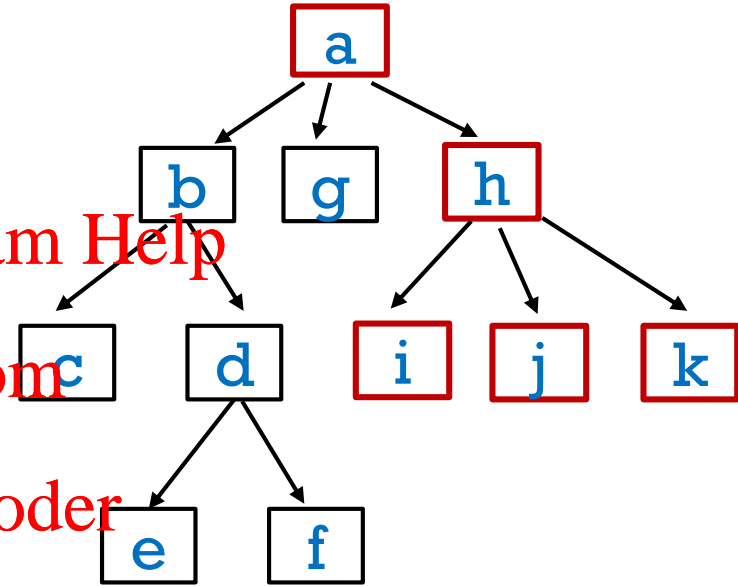
TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
  initialize empty stack s  
  s.push(root)  
  while s is not empty {  
    cur = s.pop()  
    visit cur  
    for each child of cur  
      s.push(child)  
  }  
}
```



TREE TRAVERSAL – WITH A STACK

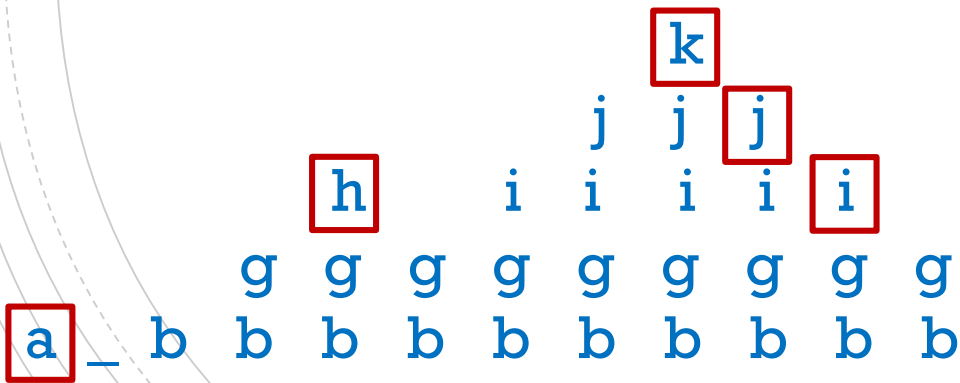
```
treeTraversalUsingStack(root) {
    initialize empty stack s
    s.push(root)
    while s is not empty {
        cur = s.pop()
        visit cur
        for each child of cur
            s.push(child)
    }
}
```



Assignment Project Exam Help

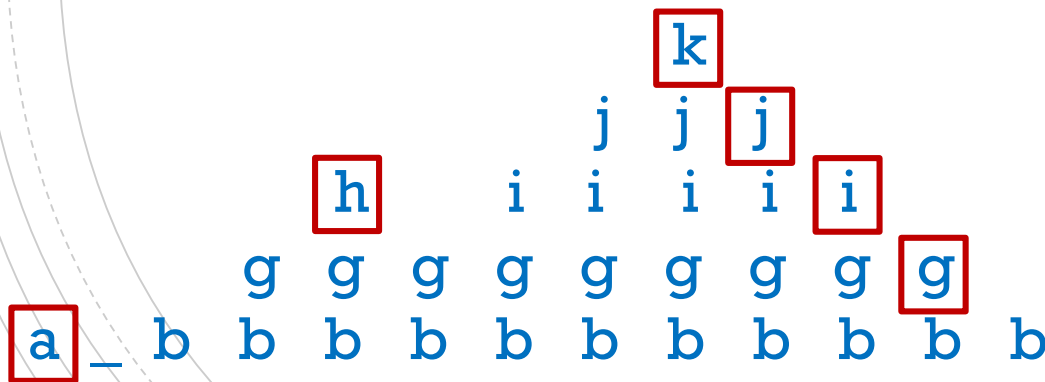
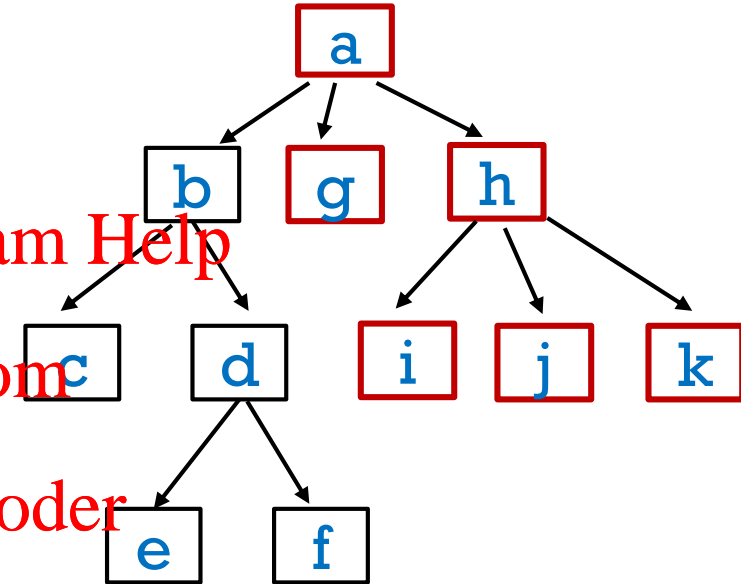
<https://powcoder.com>

Add WeChat powcoder



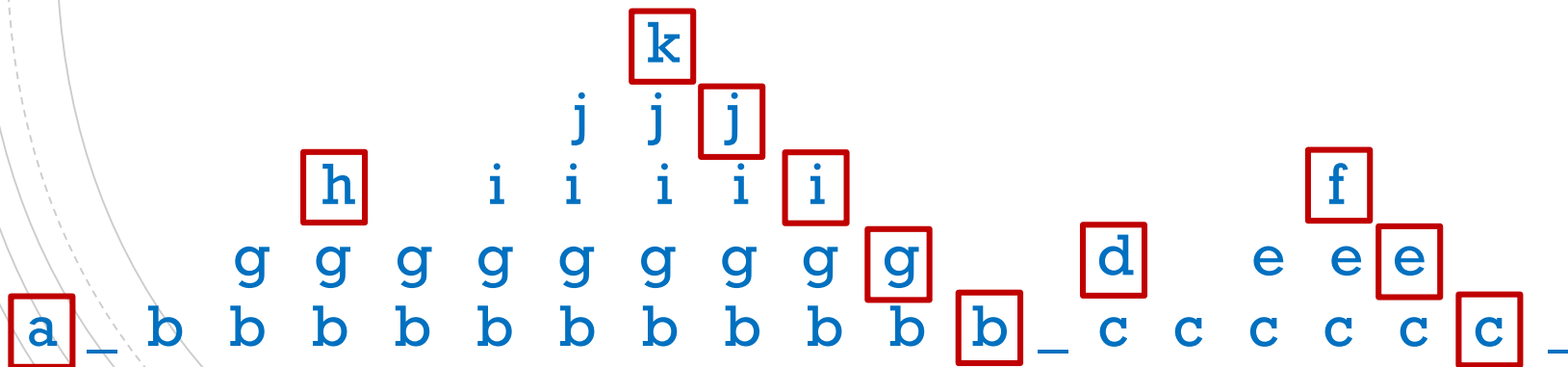
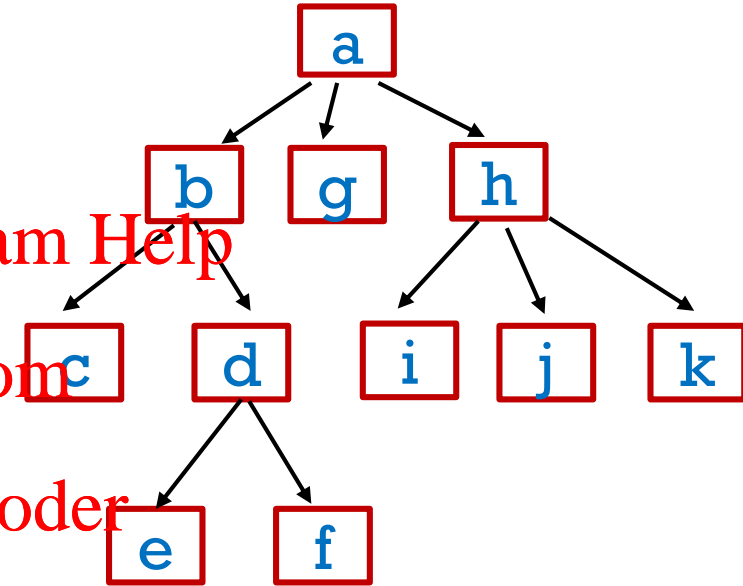
TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



TREE TRAVERSAL – WITH A STACK

```
treeTraversalUsingStack(root) {  
  initialize empty stack s  
  s.push(root)  
  while s is not empty {  
    cur = s.pop()  
    visit cur  
    for each child of cur  
      s.push(child)  
  }  
}
```



TREE TRAVERSAL – WITH A STACK

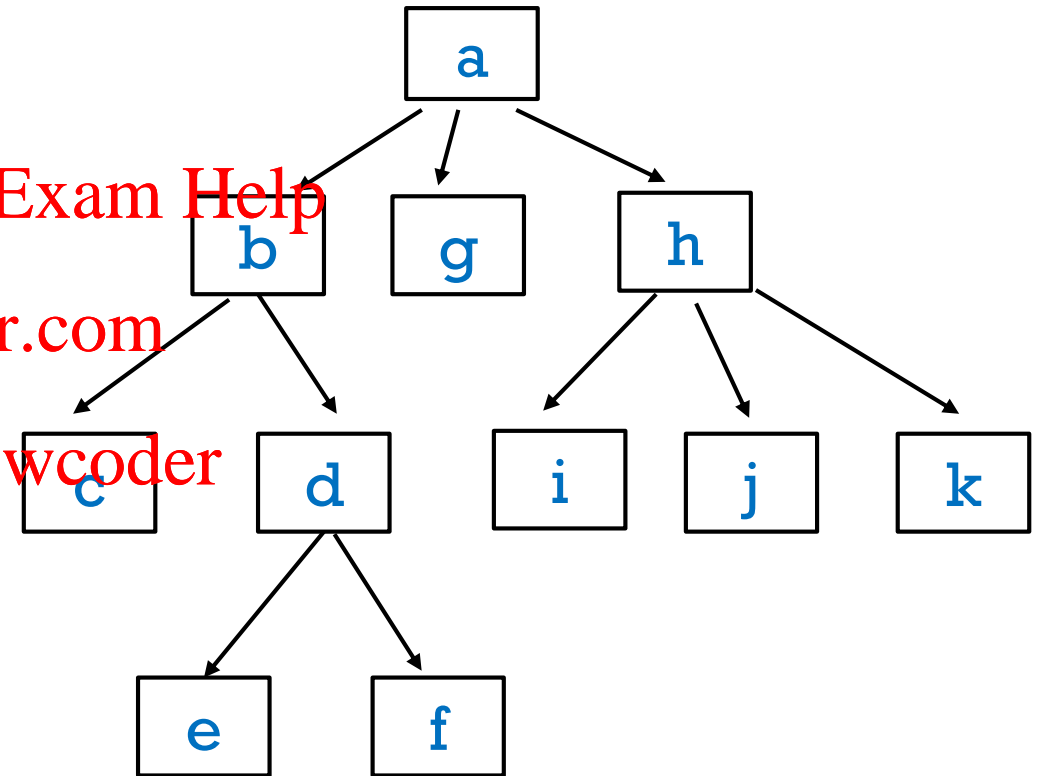
Q: Is it depth first?

A: Yes, but it visits the children “from right to left”

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Recursive preorder: abcdefghijk

Recursive postorder: cefdbgijkha

Non-recursive (stack): **ahkjigbdfec**

TREE TRAVERSAL – WITH A STACK

Q: Is it preorder or postorder?

A: It's preorder.

Q: Would moving the visit change that?


A: No... why?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```



WHAT IF WE USED A QUEUE INSTEAD?

```
treeTraversalUsingStack(root) {  
    initialize empty stack s  
    s.push(root)  
    while s is not empty {  
        cur = s.pop()  
        visit cur  
        for each child of cur  
            s.push(child)  
    }  
}
```

```
treeTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while q is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```

Assignment Project Exam Help

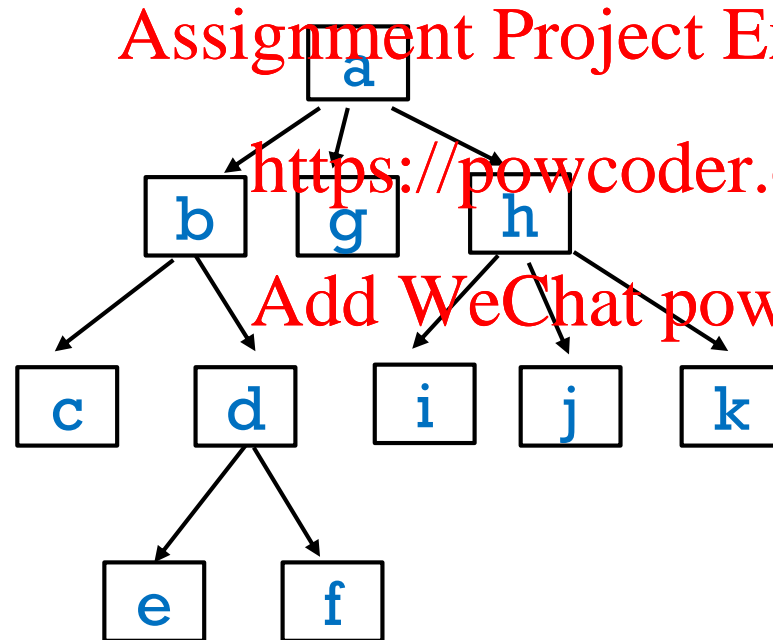
<https://powcoder.com>

Add WeChat powcoder

WHAT IF WE USED A QUEUE INSTEAD?

Queue state at start of the while loop

a



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

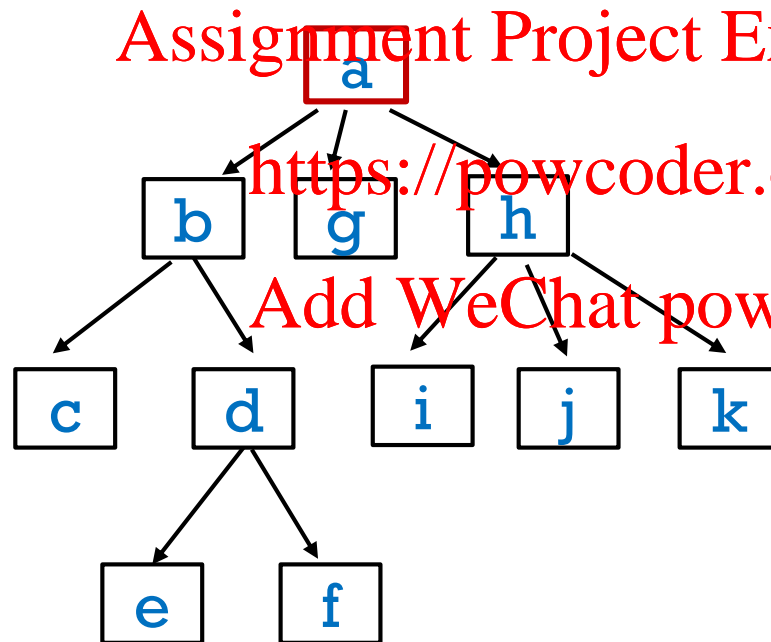
```
preOrderTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while s is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```

WHAT IF WE USED A QUEUE INSTEAD?

Queue state at start of the while loop

a

b g h



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
preOrderTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while s is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```

WHAT IF WE USED A QUEUE INSTEAD?

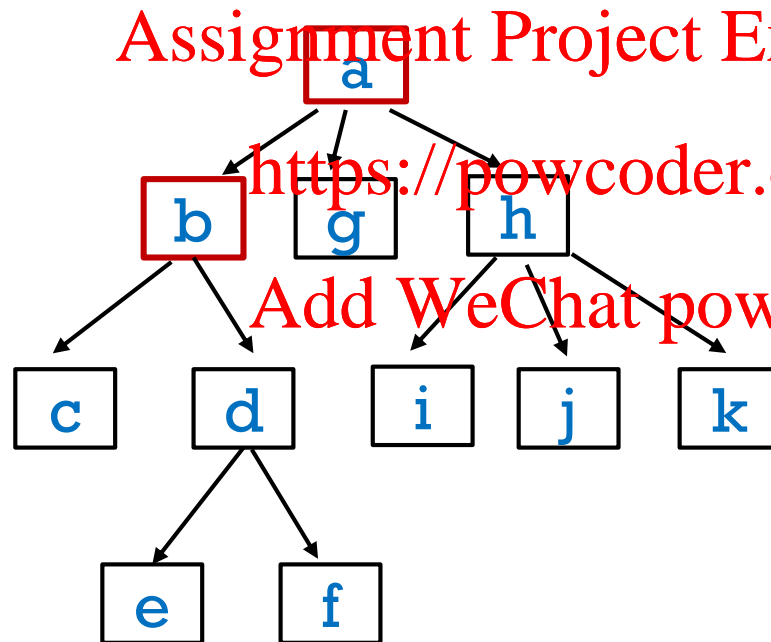
Queue state at start of the while loop

a

b g h

g h

g h c d



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
preOrderTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while s is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```

WHAT IF WE USED A QUEUE INSTEAD?

Queue state at start of the while loop

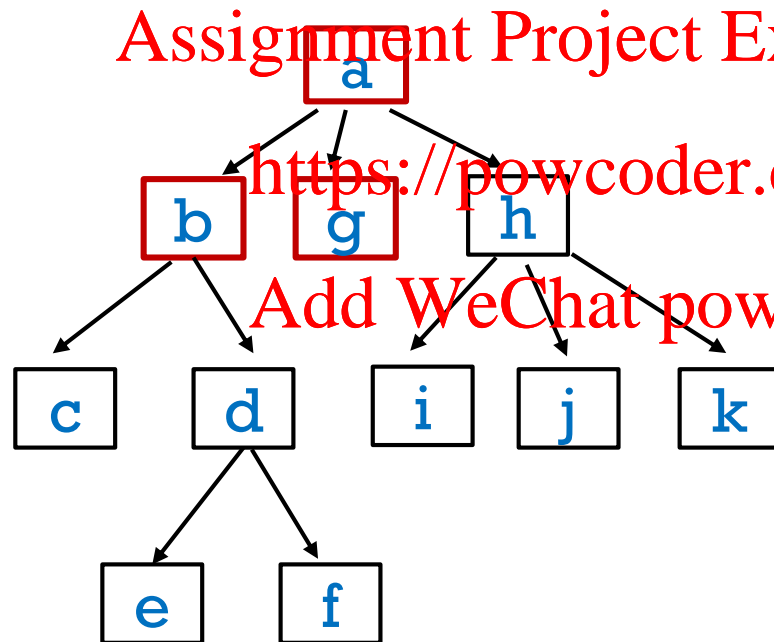
a

b g h

g h

g h c d

h c d



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
preOrderTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while s is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```


WHAT IF WE USED A QUEUE INSTEAD?

Queue state at start of the while loop

a

b g h

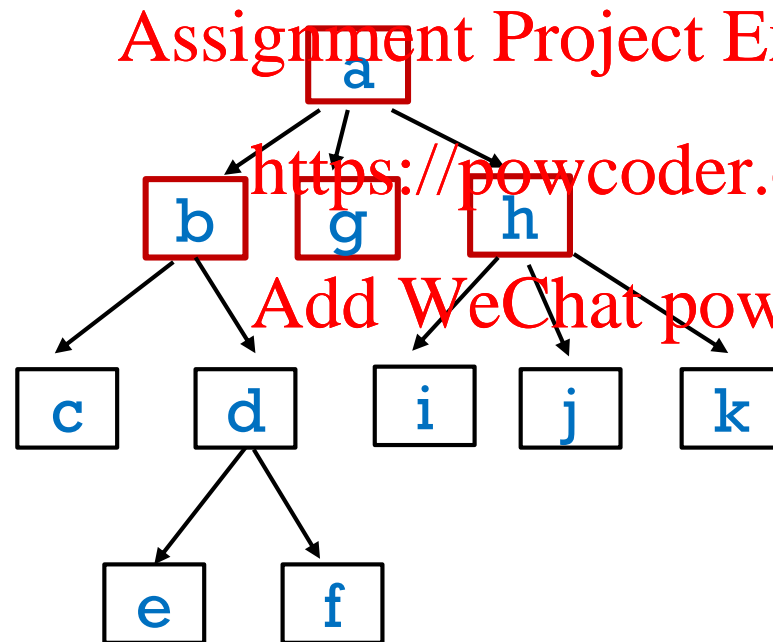
g h

g h c d

h c d

c d

c d i j k



```
preOrderTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while s is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```

WHAT IF WE USED A QUEUE INSTEAD?

Queue state at start of the while loop

a

b g h

g h

g h c d

h c d

c d

c d i j k

d i j k

i j k

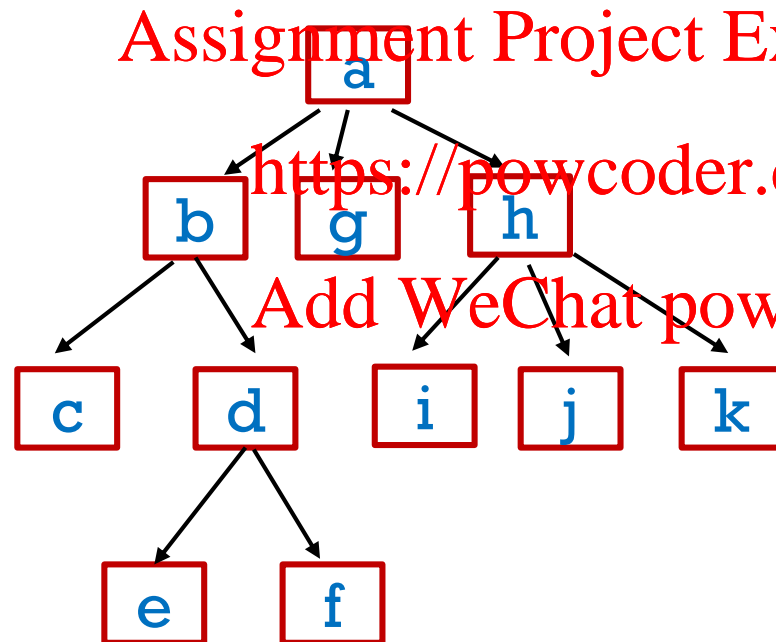
i j k e f

j k e f

k e f

e f

f



```
preOrderTraversalUsingQueue(root) {  
    initialize empty queue q  
    q.enqueue(root)  
    while s is not empty {  
        cur = q.dequeue()  
        visit cur  
        for each child of cur  
            q.enqueue(child)  
    }  
}
```

BREADTH FIRST TRAVERSAL

For each level i

visit all nodes at level i

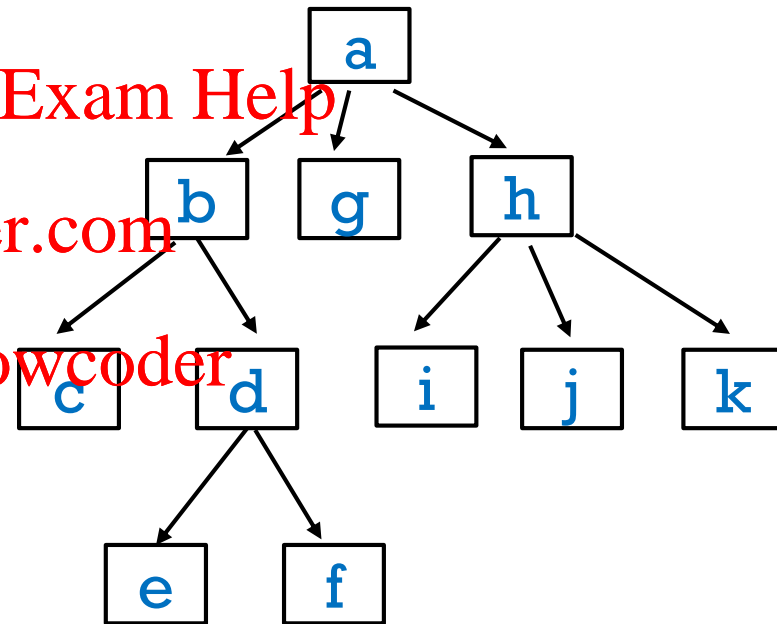
Order visited:

abghcdijkef

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



IMPLEMENTATION DETAILS

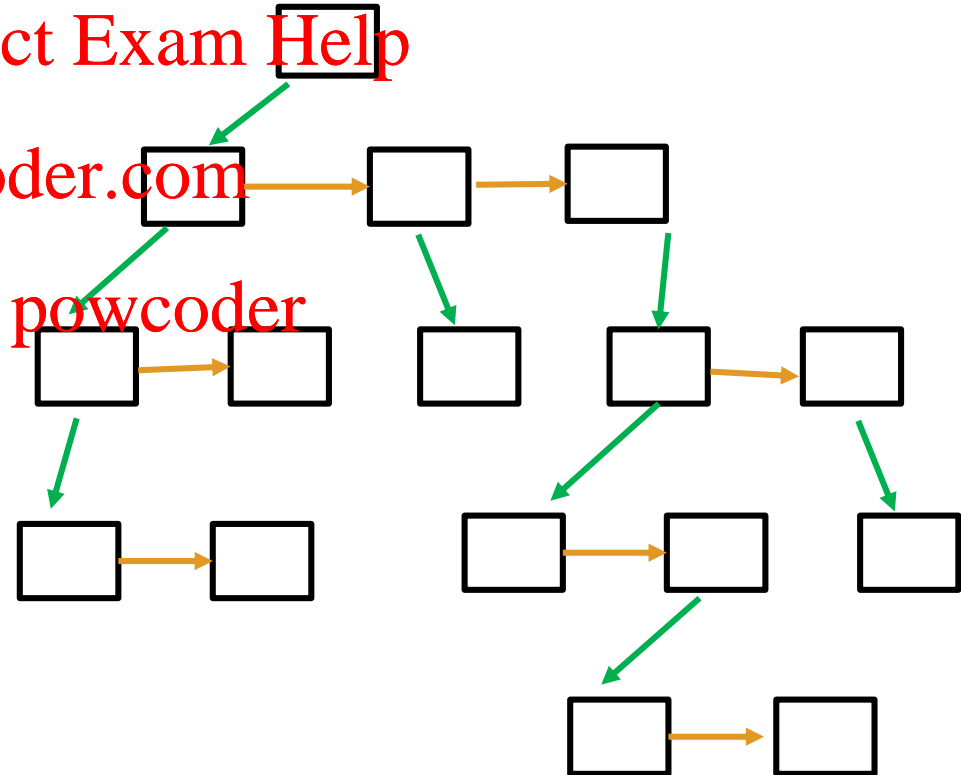
Recall the “**first child**, **next sibling**” implementation

```
class Tree<T>{  
    TreeNode<T> root;  
    :  
  
    class TreeNode<T>{  
        T element;  
        TreeNode<T> firstChild;  
        TreeNode<T> nextSibling;  
        :  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



IMPLEMENTATION DETAILS

Recall the “**first child**, **next sibling**” implementation

Then when we write

```
for each child {
```

```
    :
```

```
}
```

it means

```
child = cur.firstChild
```

```
while(child !=null) {
```

```
    :
```

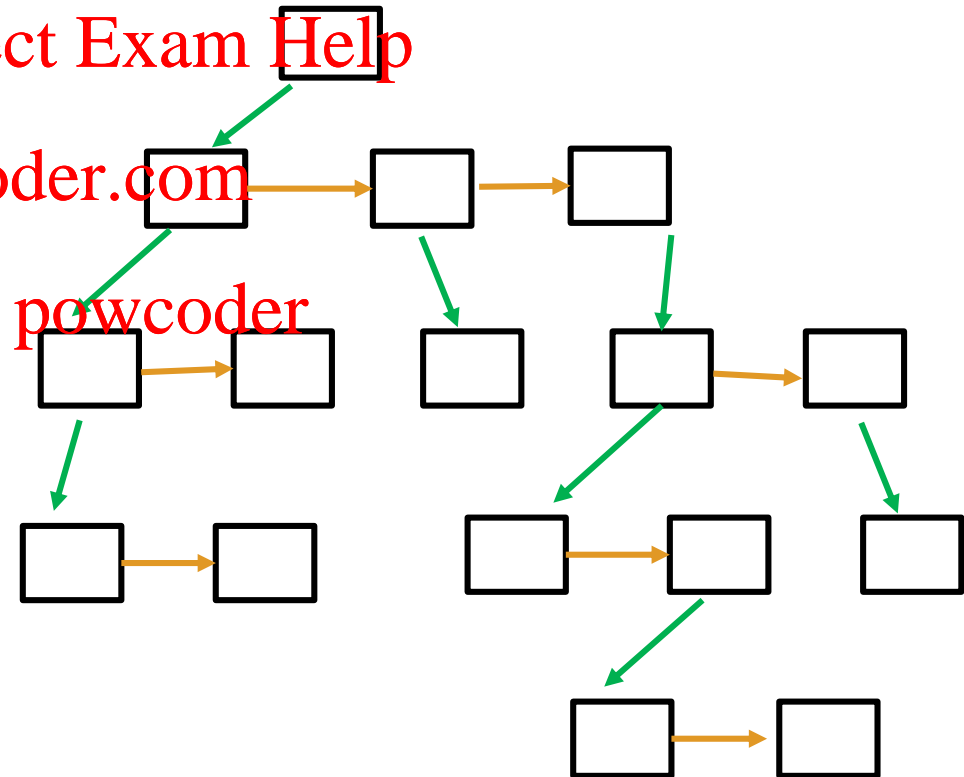
```
    child = child.nextSibling
```

```
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder





Coming Soon

Assignment Project Exam Help

In the next video:

■ **Binary Trees**

<https://powcoder.com>

Add WeChat powcoder