COMP 250

INTRODUC TER SCIENCE

Week 11-2: Tra

Giulia Alberini, Fall 2020

Slides adapted from Michael Langer's

WHAT ARE WE GOING TO DO IN THIS VIDEO?

- Tree traversals Assignment Project Exam Help

  - Depth first VS

    https://eduassistpro.github.io/

  - Recursive and Non-recursive   k or with queue)

    Add WeChat edu_assist_pro

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

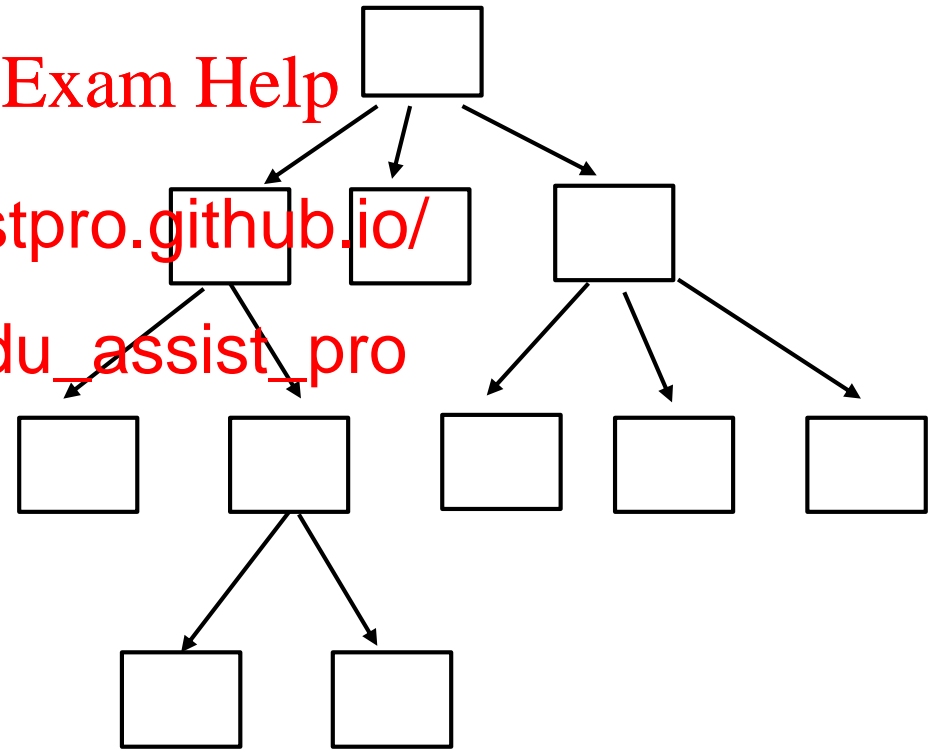How to visit (enumerate, iterate through, traverse…) all the nodes of a tree ?
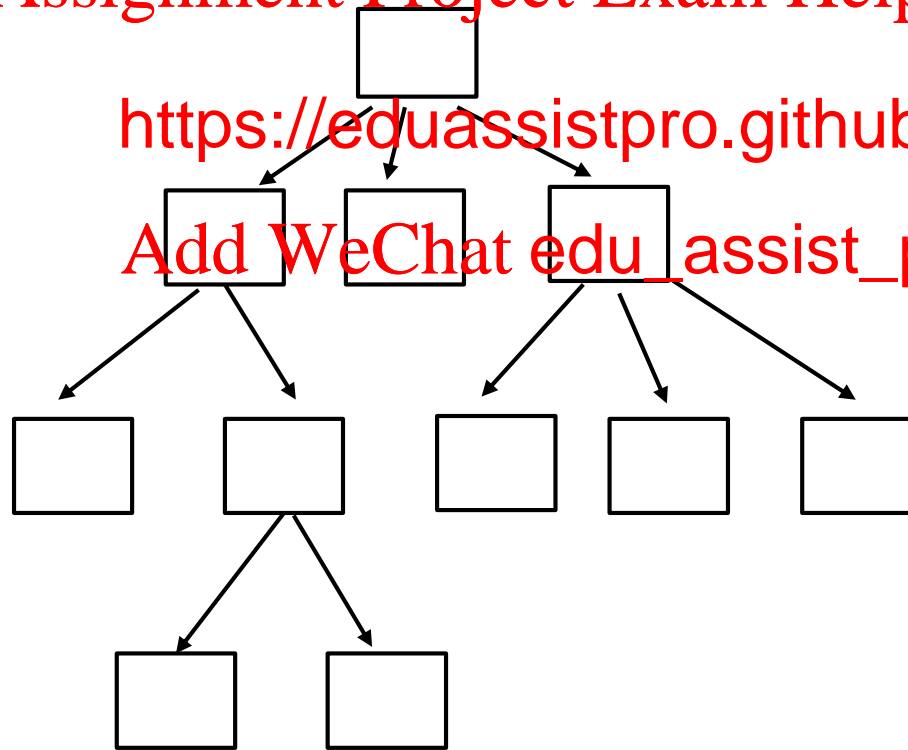
```
depthFirst (root){
    if (root is not em
        visit root
        for each child of root
            depthfirst( child )
    }
}
```

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro
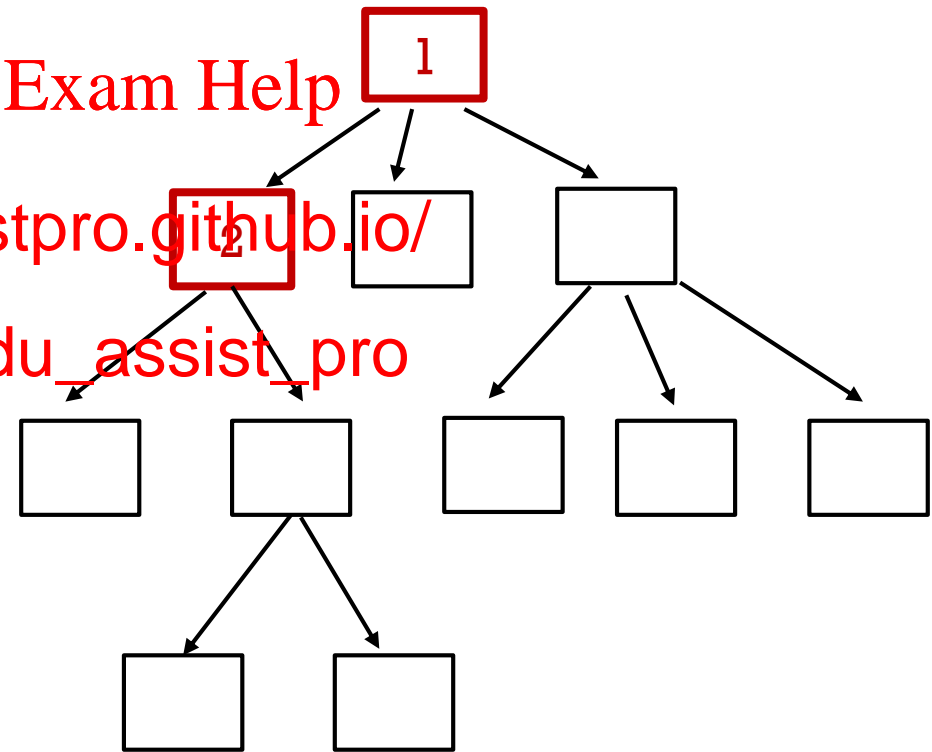
"preorder" traversal: visit the root before the children
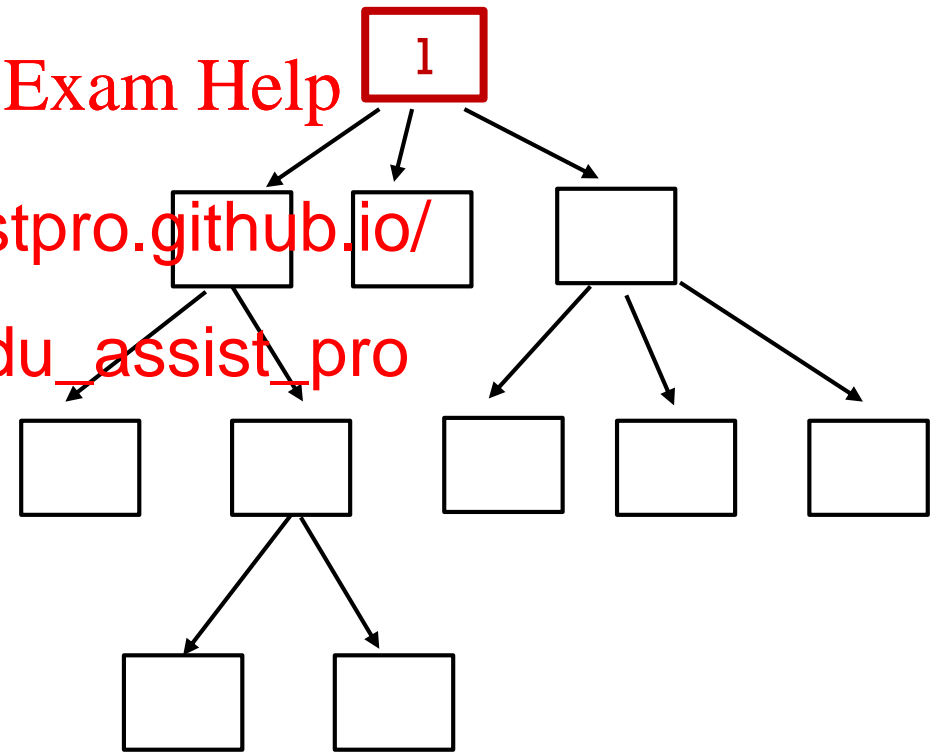
```
depthFirst (root){
   if (root is not em
      visit root
      for each child of root
         depthfirst( child )
   }
}
```

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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

```
depthFirst (root){
    if (root is not em
        visit root
        for each child of root
            depthfirst( child )
    }
}
```
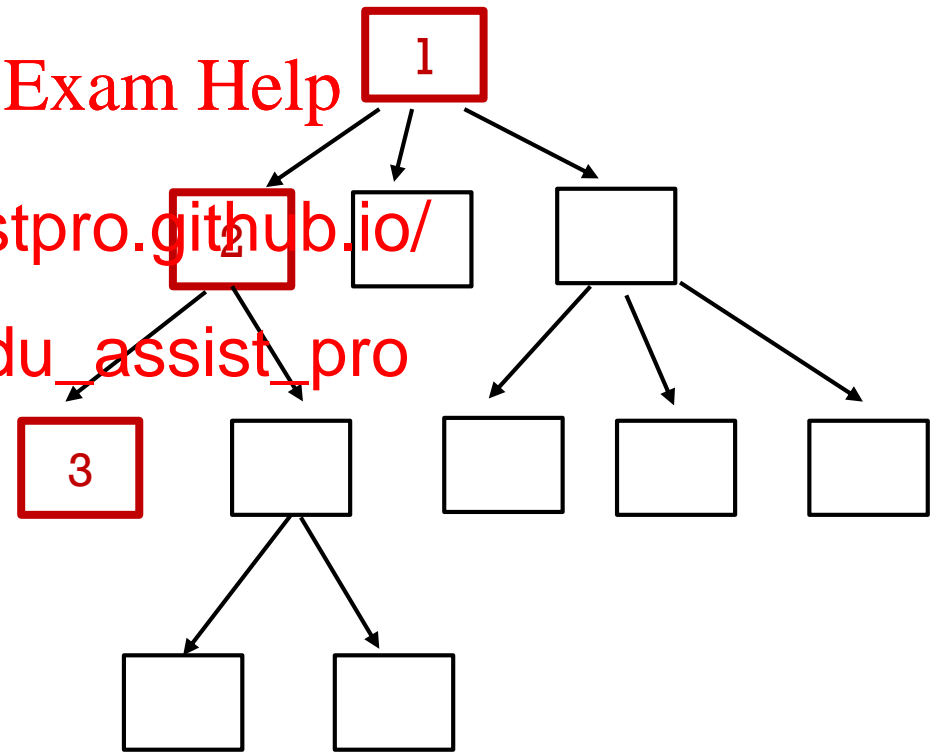
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
depthFirst (root){
    if (root is not em
        visit root
        for each child of root
            depthfirst( child )
    }
}
```
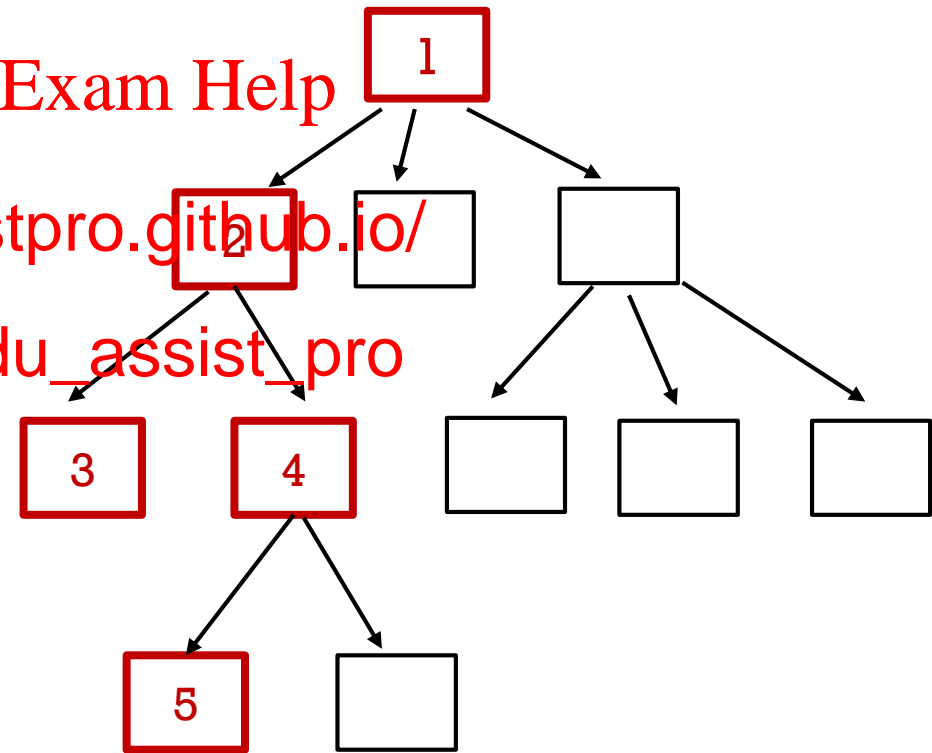
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
depthFirst (root){
    if (root is not em
        visit root
        for each child of root
            depthfirst( child )
    }
}
```
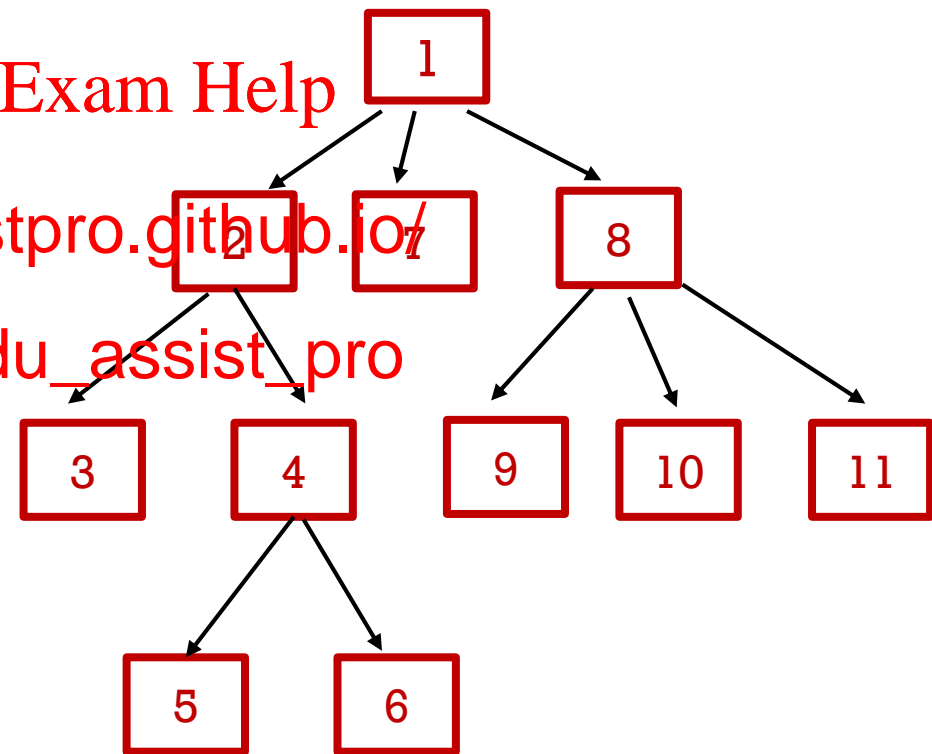
Assignment Project Exam Help
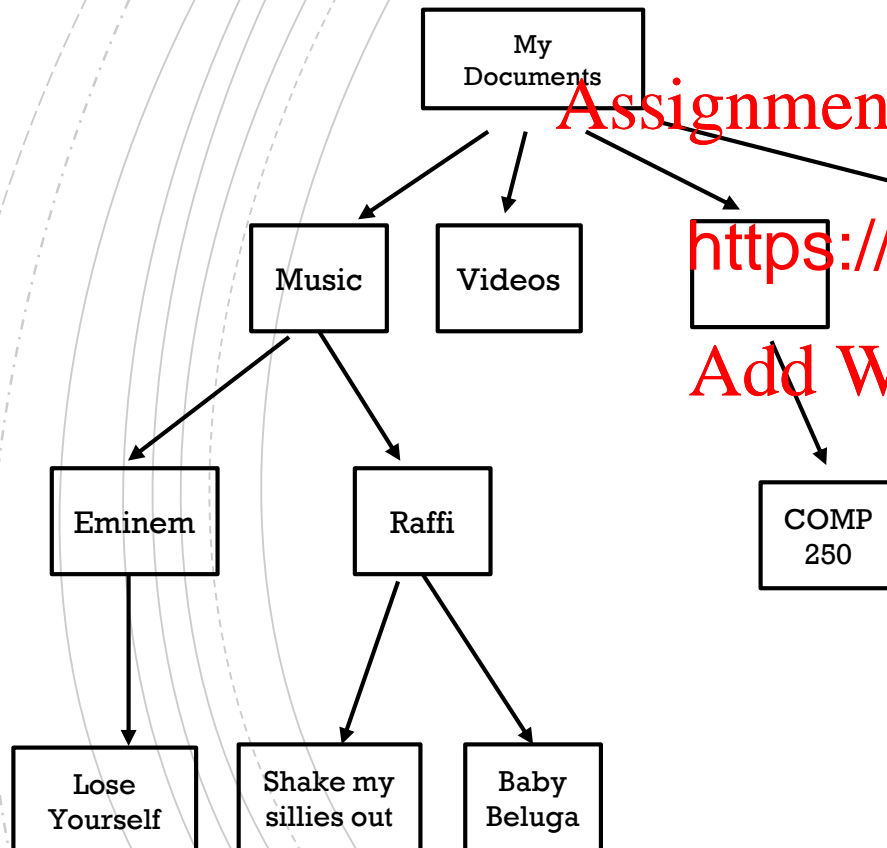
https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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)
:	(file)
Work	(directory)
COMP250	(directory)
:	
Research	(directory)
:	

"Visit" implies that you do something at that node.

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

Q: Which node is visited first?

```
depthFirst (root){
    if (root is not em    Assignment Project Exam Help
        for each child o https://eduassistpro.github.io/
            depthfirst( child )
        visit root            Add WeChat edu_assist_pro
    }
}
```

"postorder" traversal: visit
the root after the children

```
depthFirst (root){
    if (root is not em
        for each child o
            depthfirst( child )
        visit root
    }
}
```

Assignment Project Exam Help

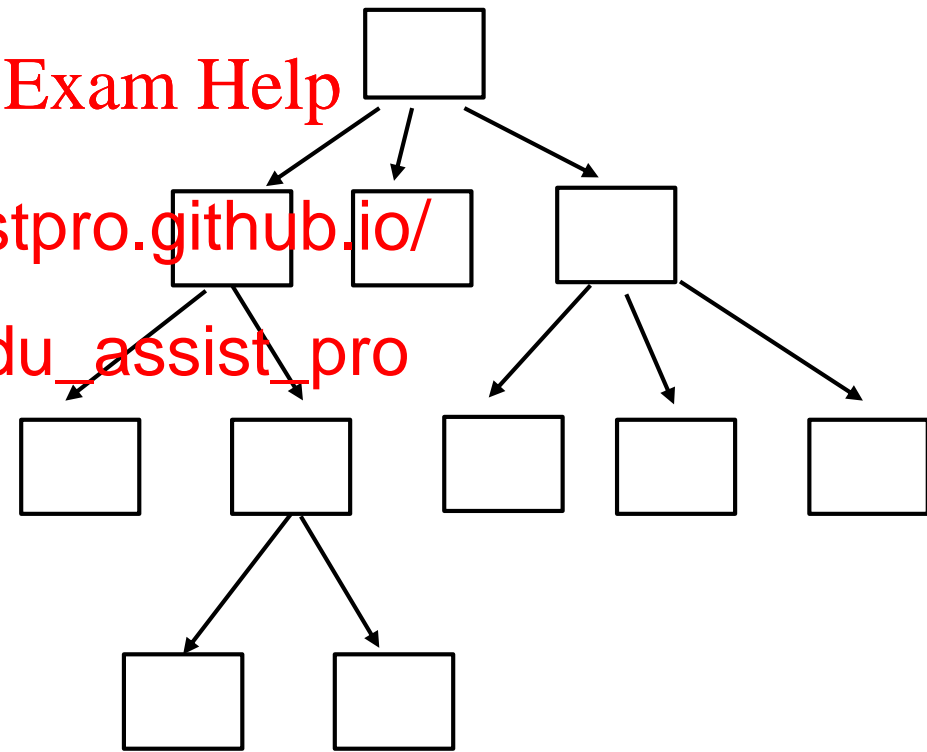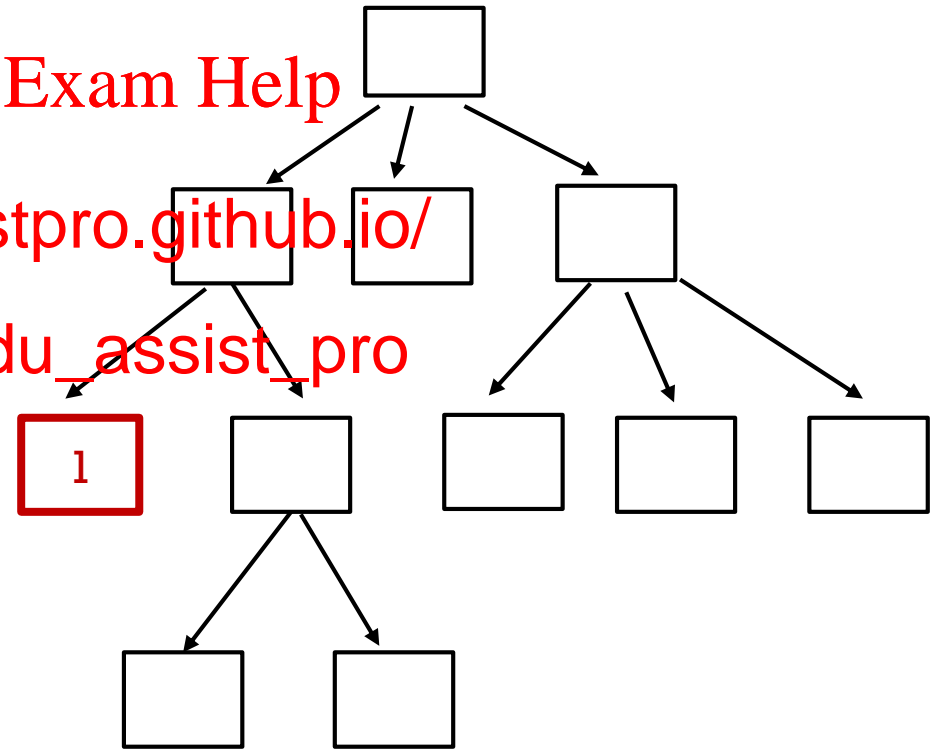https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
depthFirst (root){
    if (root is not em
        for each child o
            depthfirst( child )
        visit root
    }
}
```
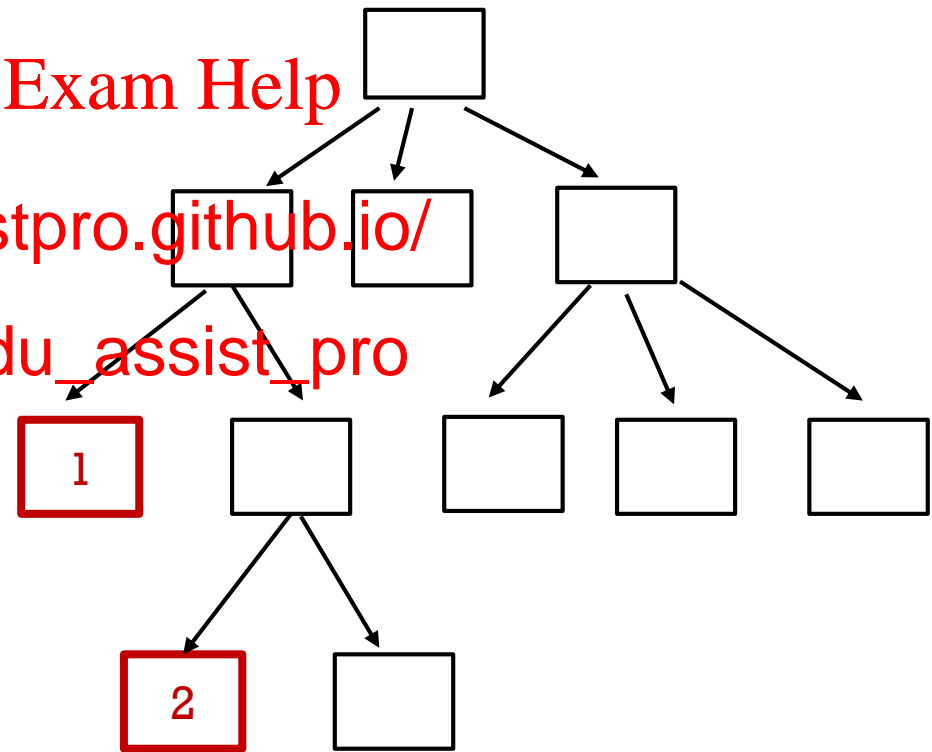
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
depthFirst (root){
    if (root is not em
        for each child o
            depthfirst( child )
        visit root
    }
}
```
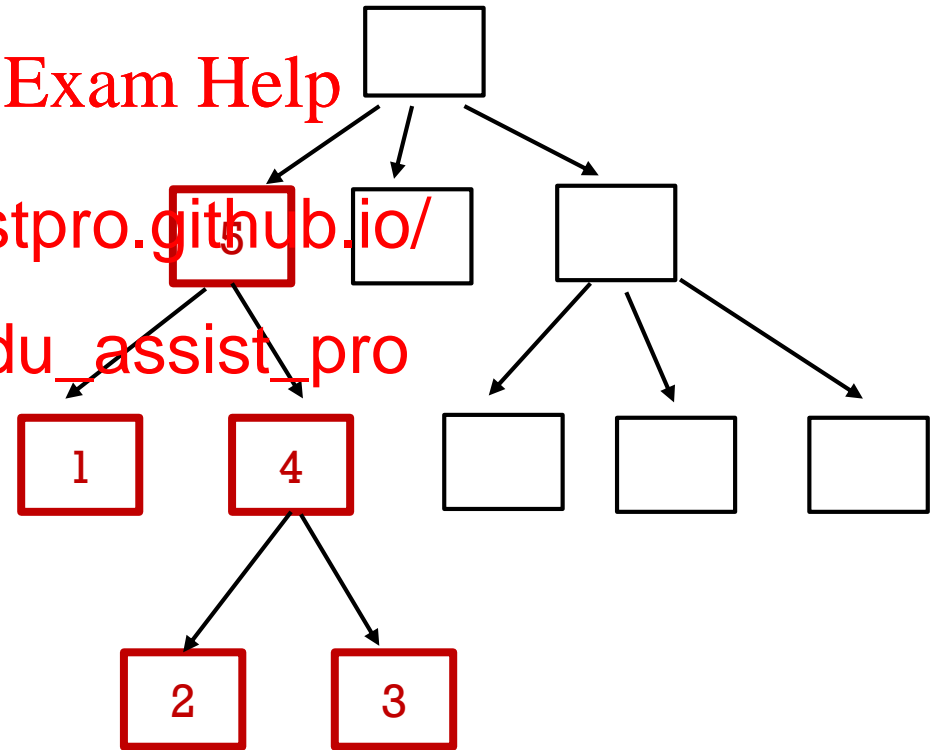
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

1   4

2   3

```
depthFirst (root){
    if (root is not em
        for each child o
            depthfirst( child )
        visit root
    }
}
```
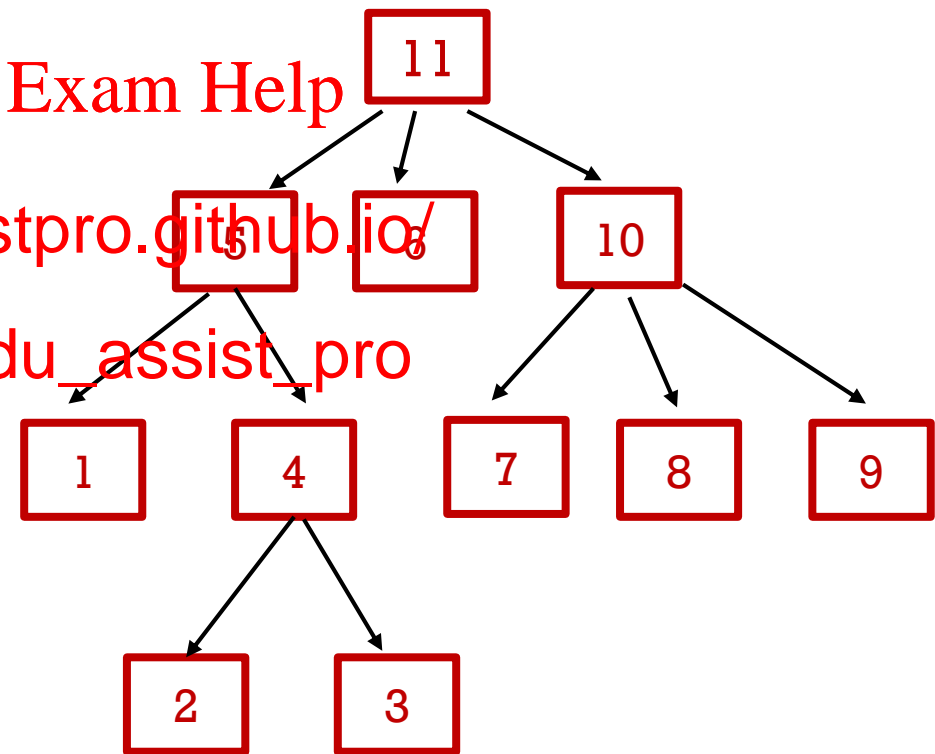
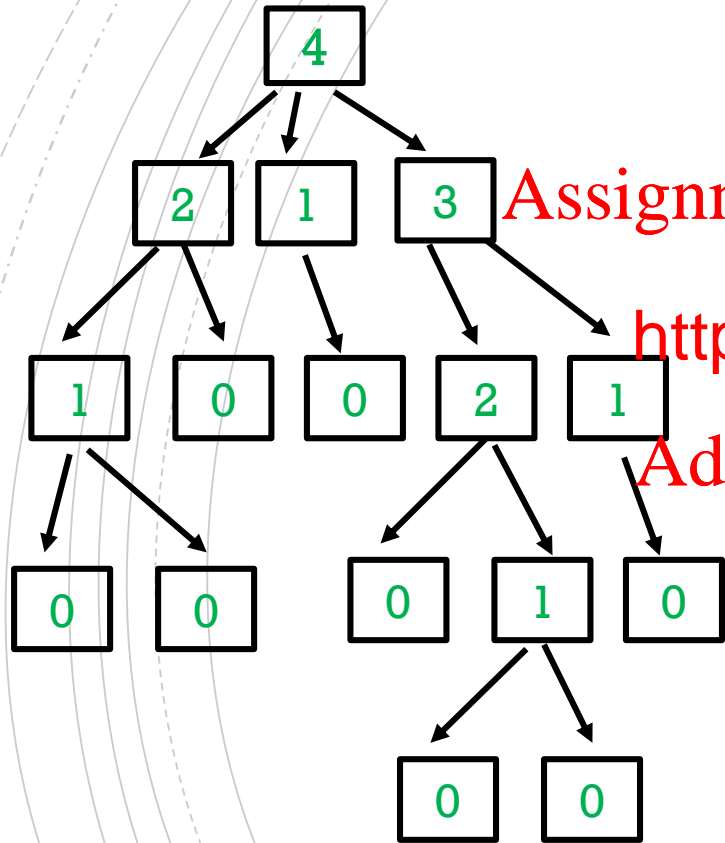Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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


```
height(v){
    if (v is a leaf)
        return 0

                   child w of v
            h = max(h, height(w))
        return 1 + h
    }
}
```

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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://eduassistpro.github.io/
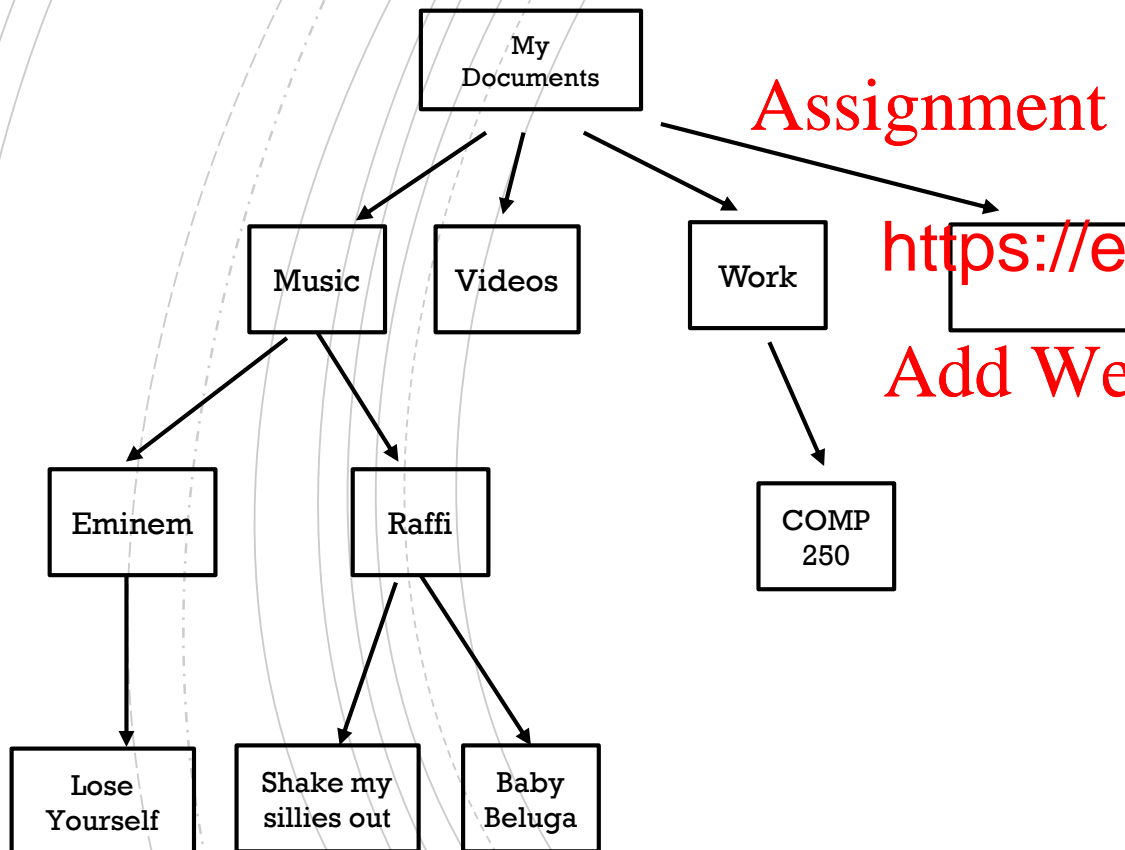
Add WeChat edu_assist_pro

```
numBytes(v){
    if (v is a leaf)
        rn number of bytes at v
        0
        ch 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 (root){
    if (root is not em
        visit root
        for each child of root
            depthfirst( child )
    }
}
```

```
depthFirst (root){
        root is not empty){
                      child of root
            depthfirst( child )
            sit root
    }
}
```
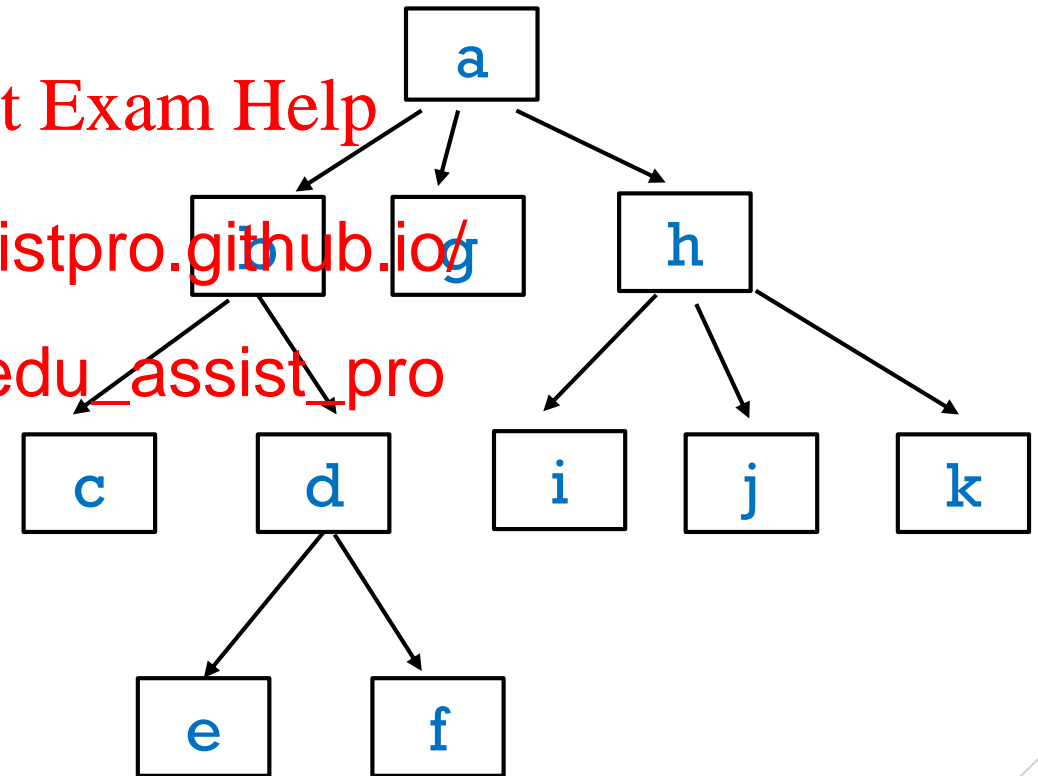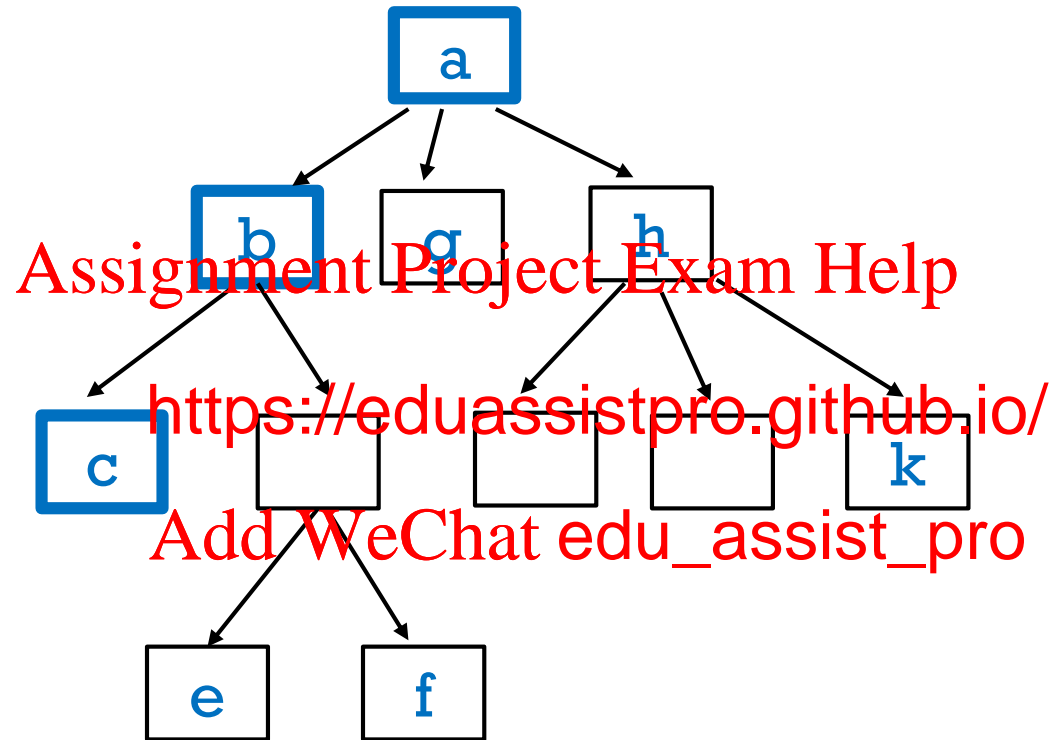
CALL SEQUENCE OF `depthFirst()`

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

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

CALL STACK FOR depthFirst(root)

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

CALL STACK FOR depthFirst(root)

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

CALL STACK FOR depthFirst(root)

a

b   g   h

Assignment Project Exam Help

c   https://eduassistpro.github.io/   k

Add WeChat edu_assist_pro

e   f

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

CALL STACK FOR depthFirst(root)

a

b   g   h
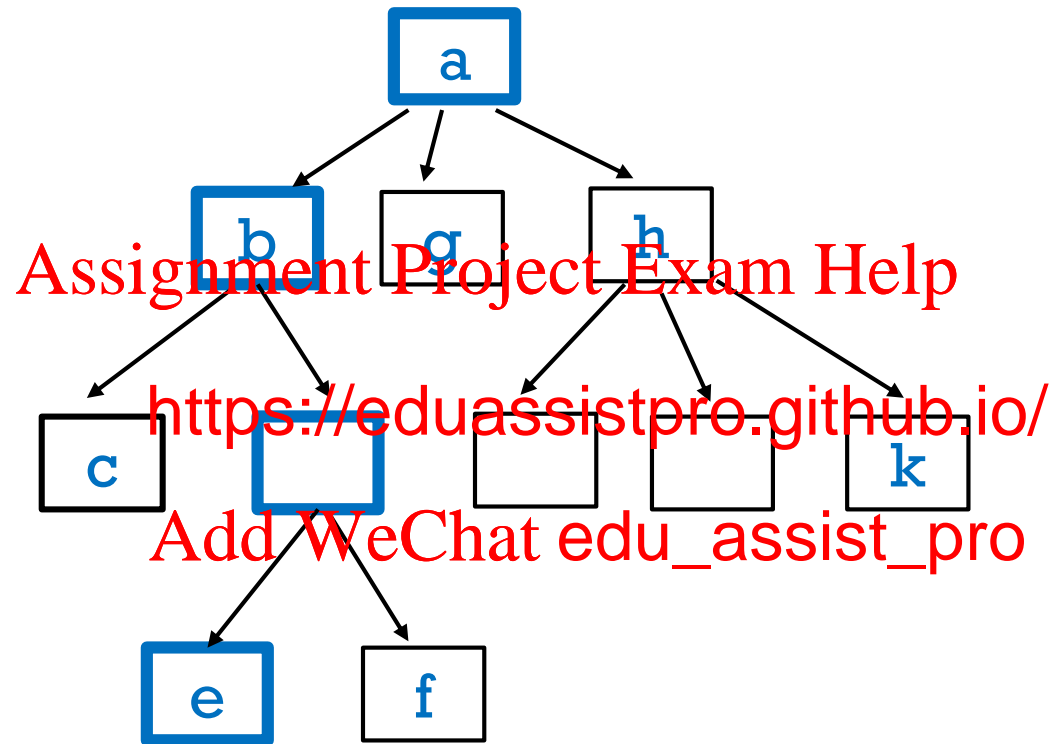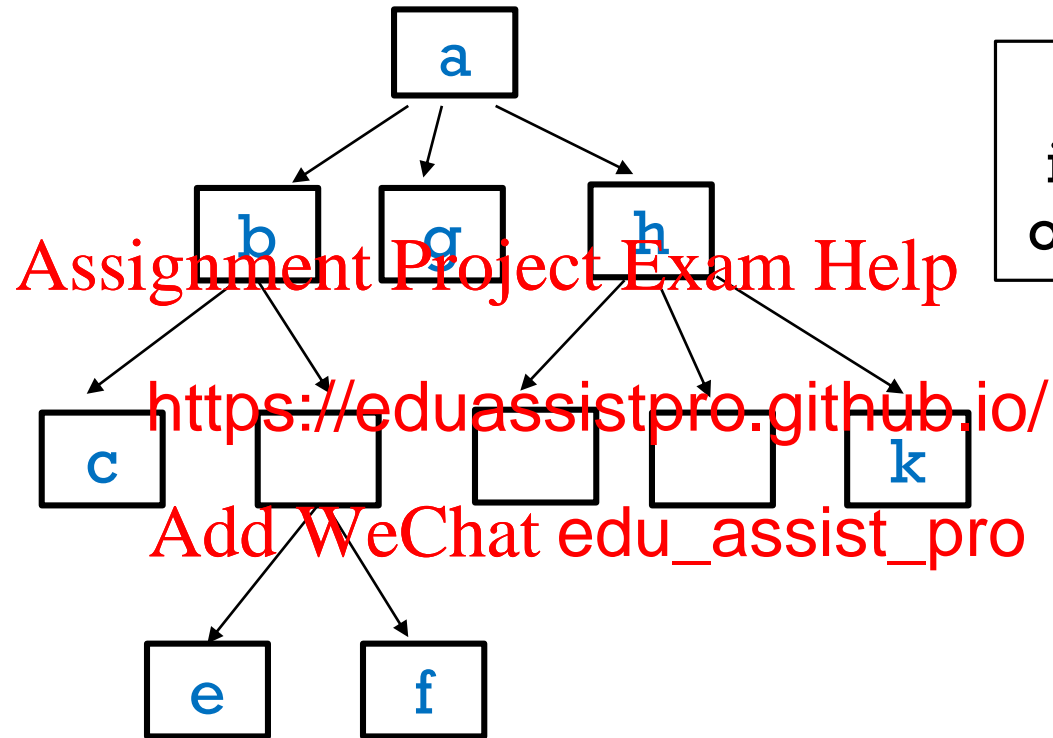
Notation: the letters indicate the call order of **depthFirst(root)**

c   k

e   f

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

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

fibonacci(247)

fibonacci(246

fibonacci(245)

fibonacci(245)

fibonacci(244)

fibonacci(244)

fibonacci(243)

fibonacci(244)   fibonacci(243)

fibonacci(243)   fibonacci(242)

Etc.
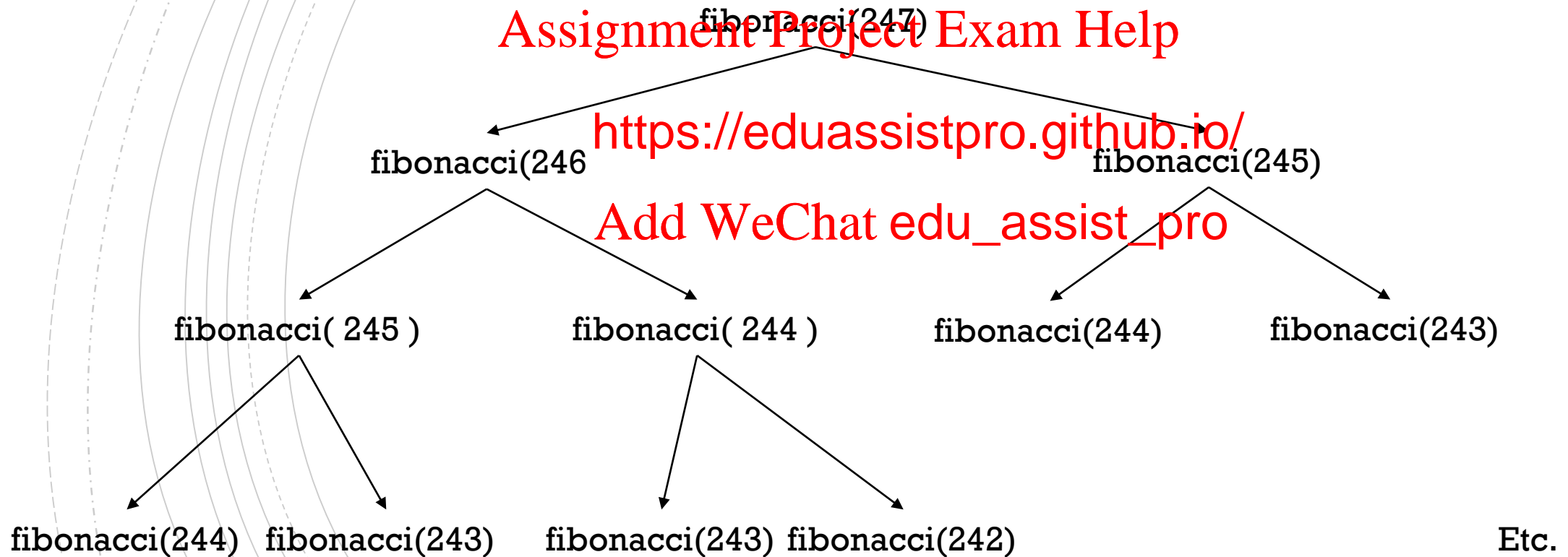
TREE TRAVERSAL IMPLEMENTATIONS

Recursive

- depth first (pre- versus post-order)

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Non-Recursive

- using a stack
- using a queue

```
treeTraversalUsingStack(root){
    initialize empty stack s
    s.push(root)




}
```

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


    }
}
```

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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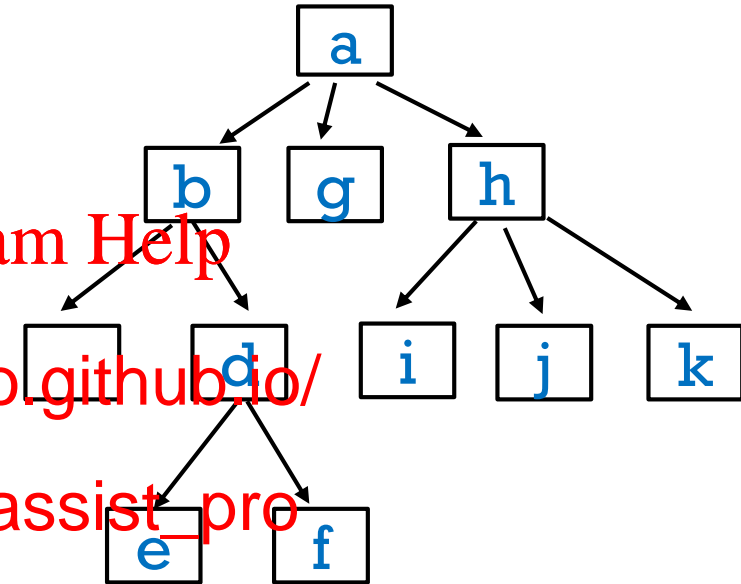
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

hat is the order in which
e nodes are visited?

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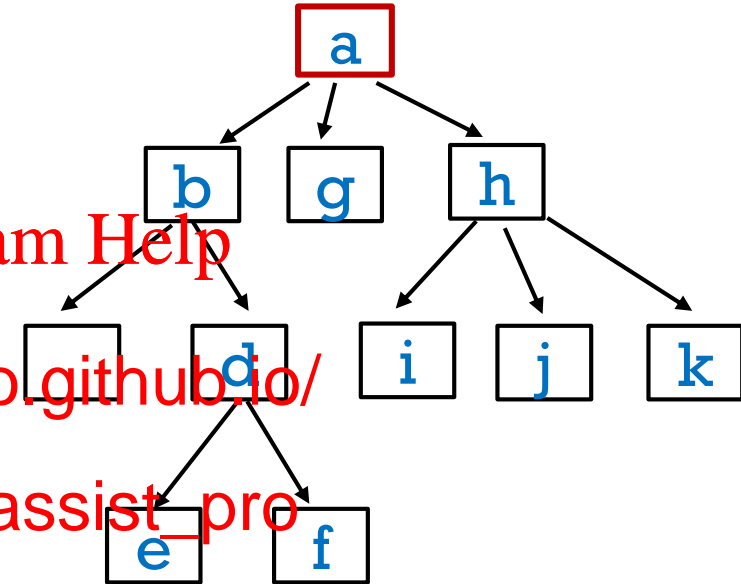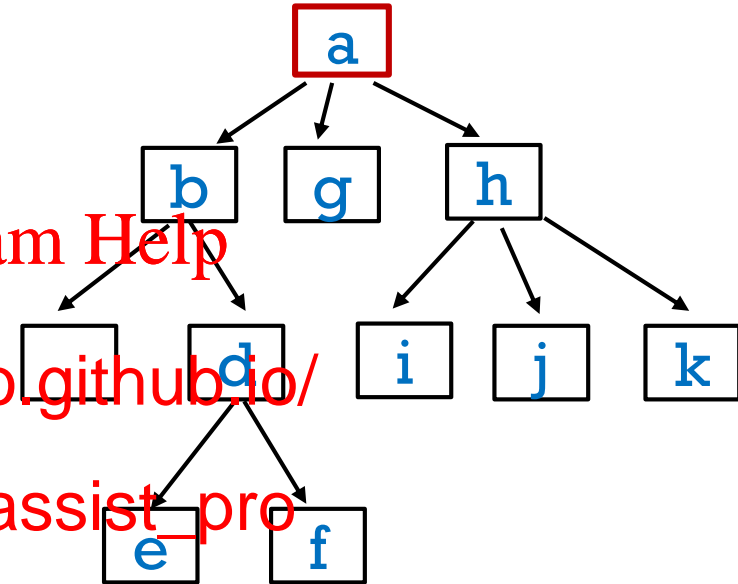
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

a

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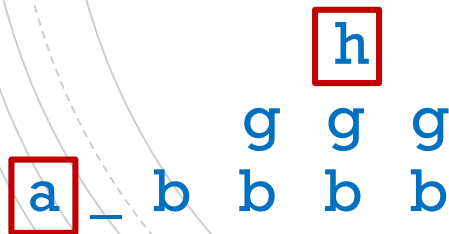
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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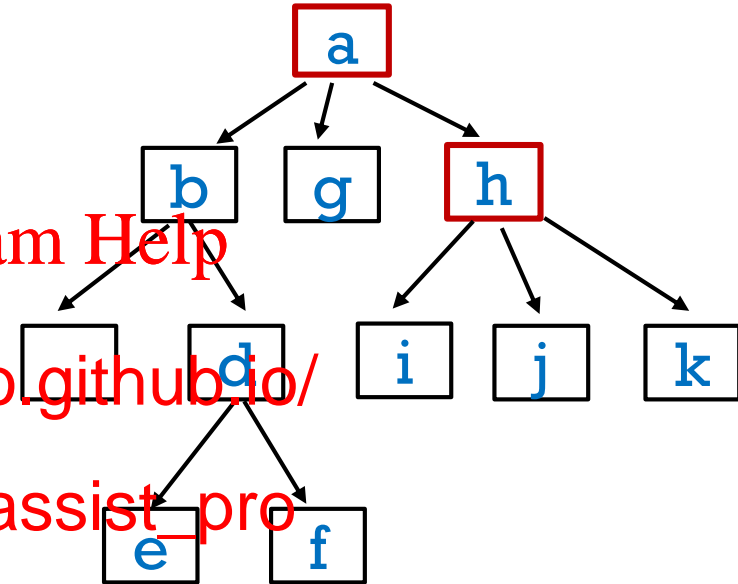
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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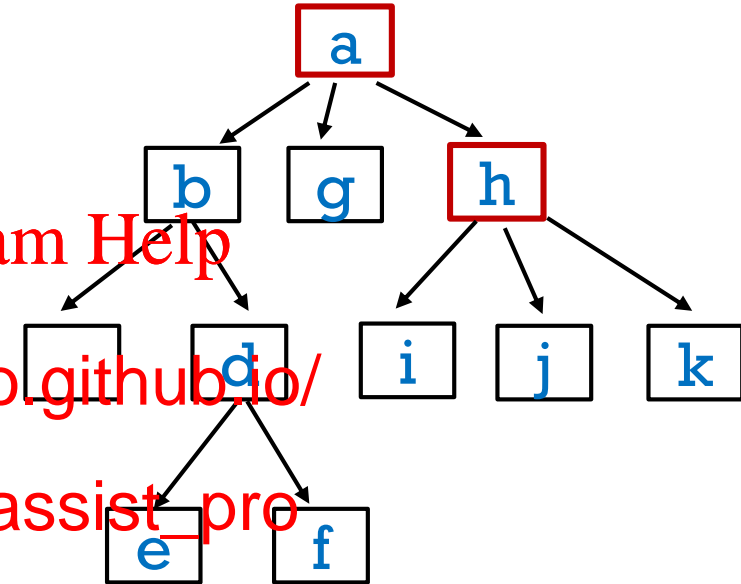
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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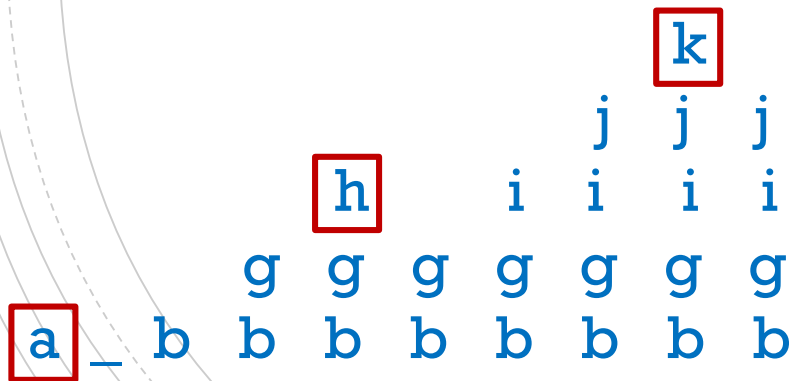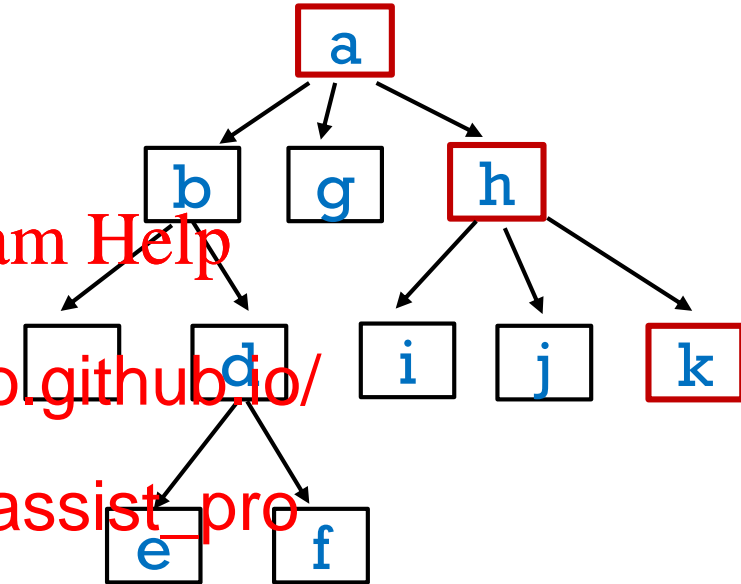
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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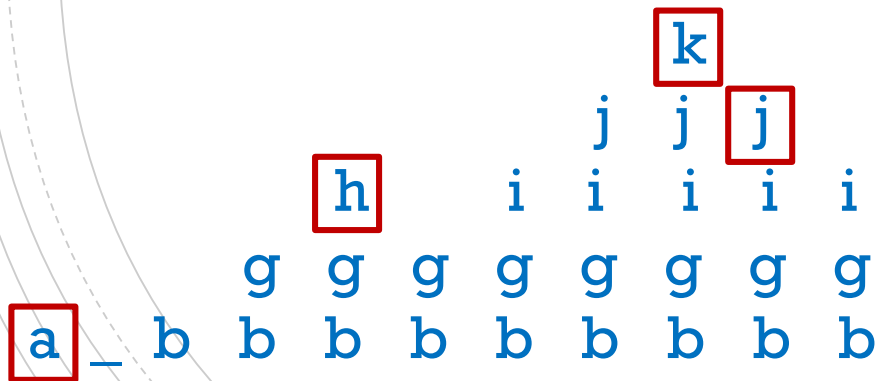
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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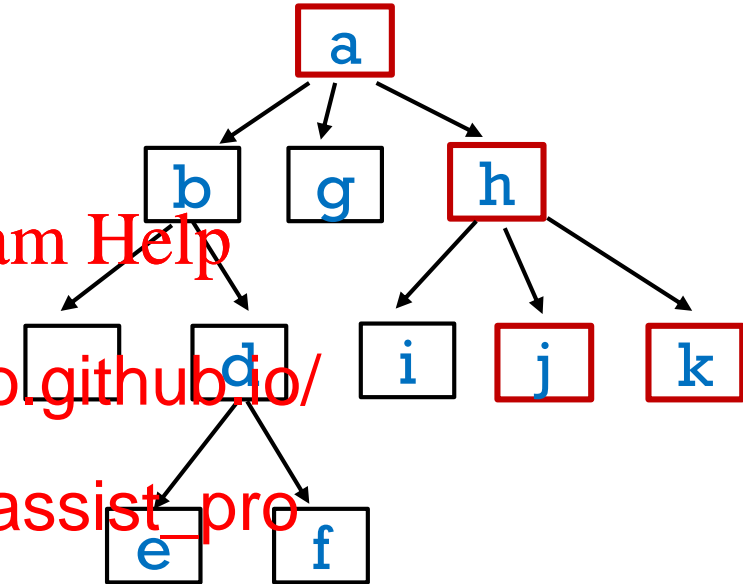
```
treeTraversalUsingStack(root){
    initialize empty stack s
    s.push(root)
    while s is not empty {
        cur = s.pop()
        visit cur
        for each child
            s.push(child)
    }
}
```
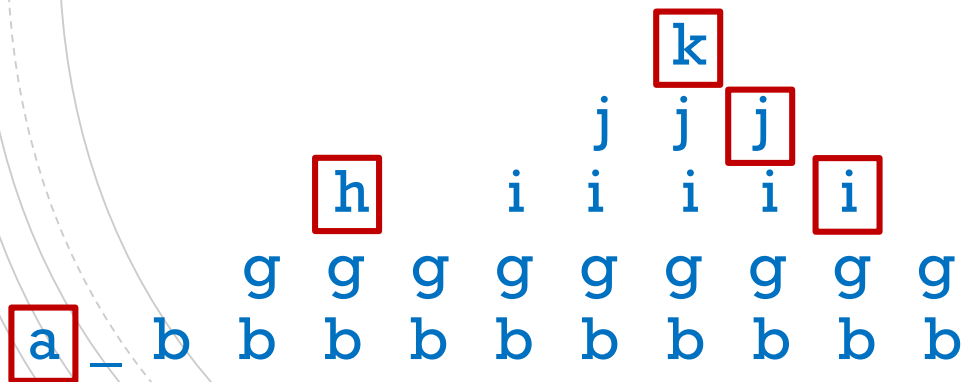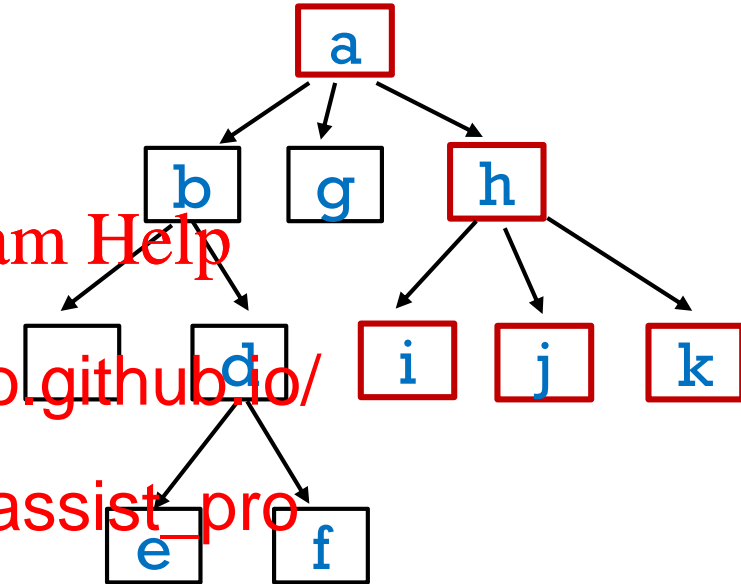
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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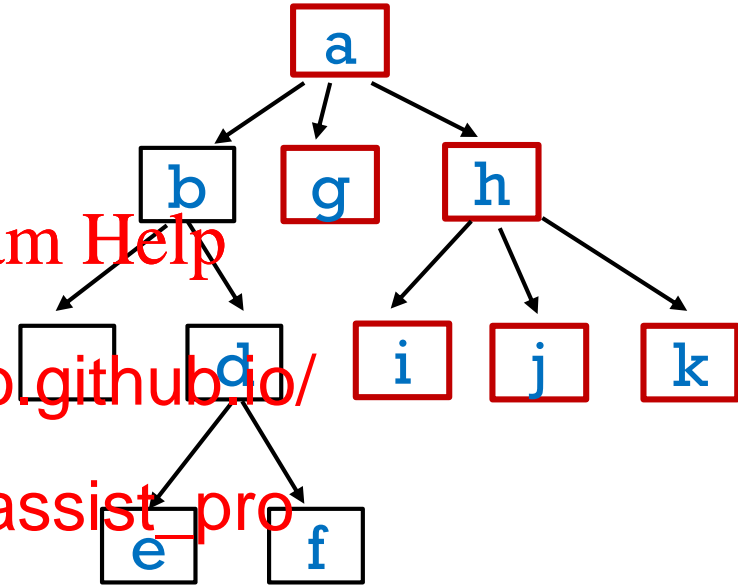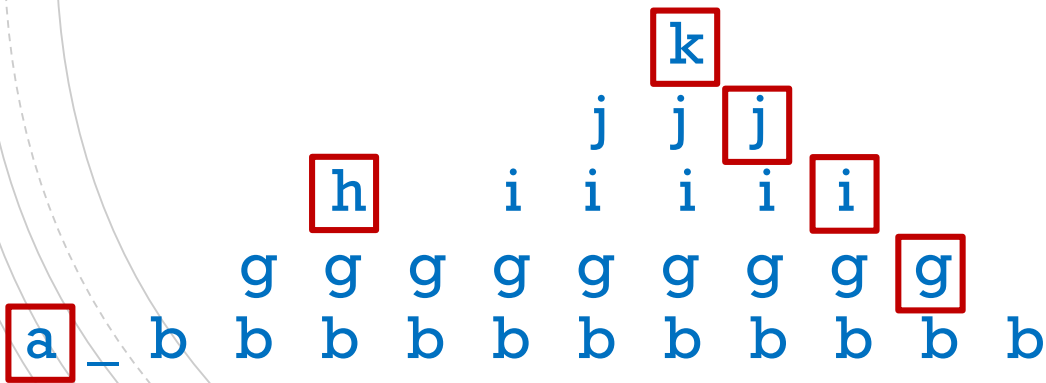
```
treeTraversalUsingStack(root){
    initialize empty stack s
    s.push(root)
    while s is not empty {
        cur = s.pop()
        visit cur
        for each child
            s.push(child)
    }
}
```
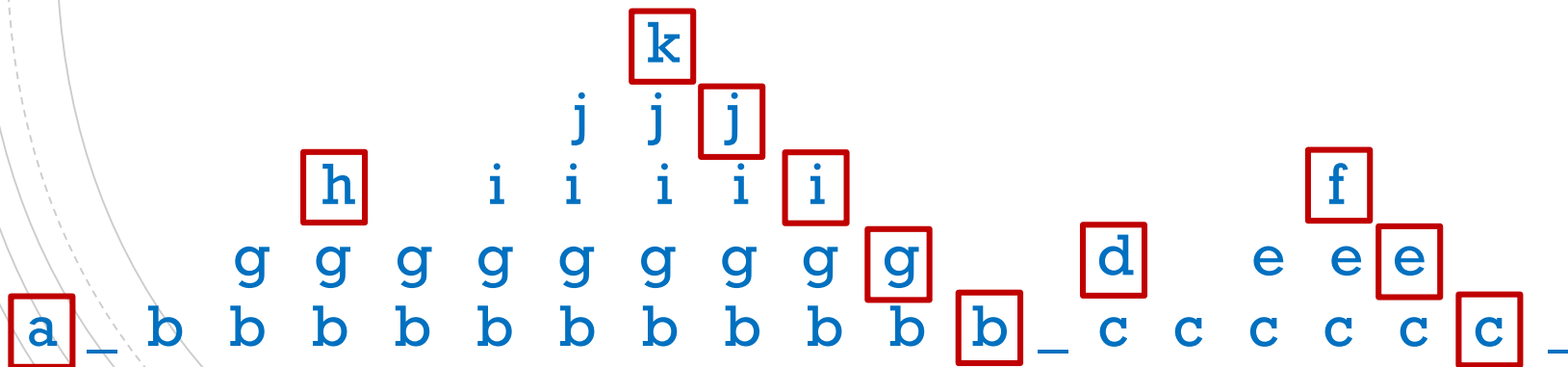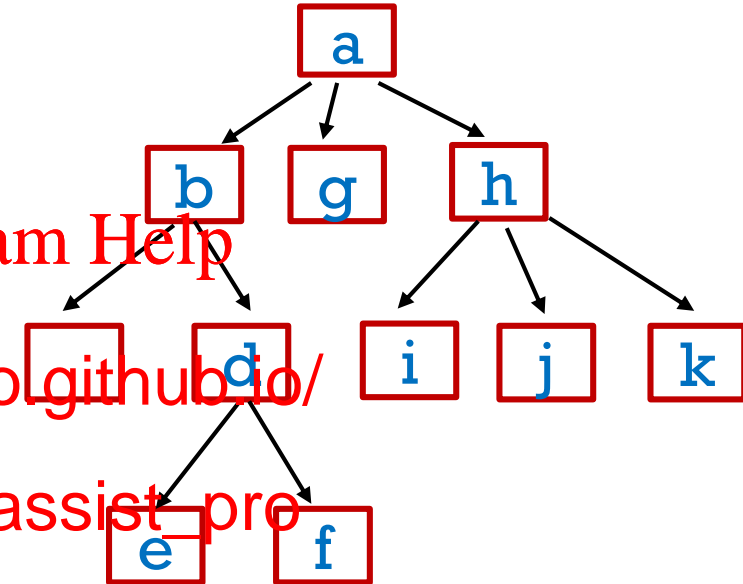
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Q: Is it depth first?

A: Yes, but it visits the chi
right to left"

Recursive preorder: abcdefghijk
Recursive postorder: cefdbgijkha

Non-recursive (stack): ahkjigbdfec
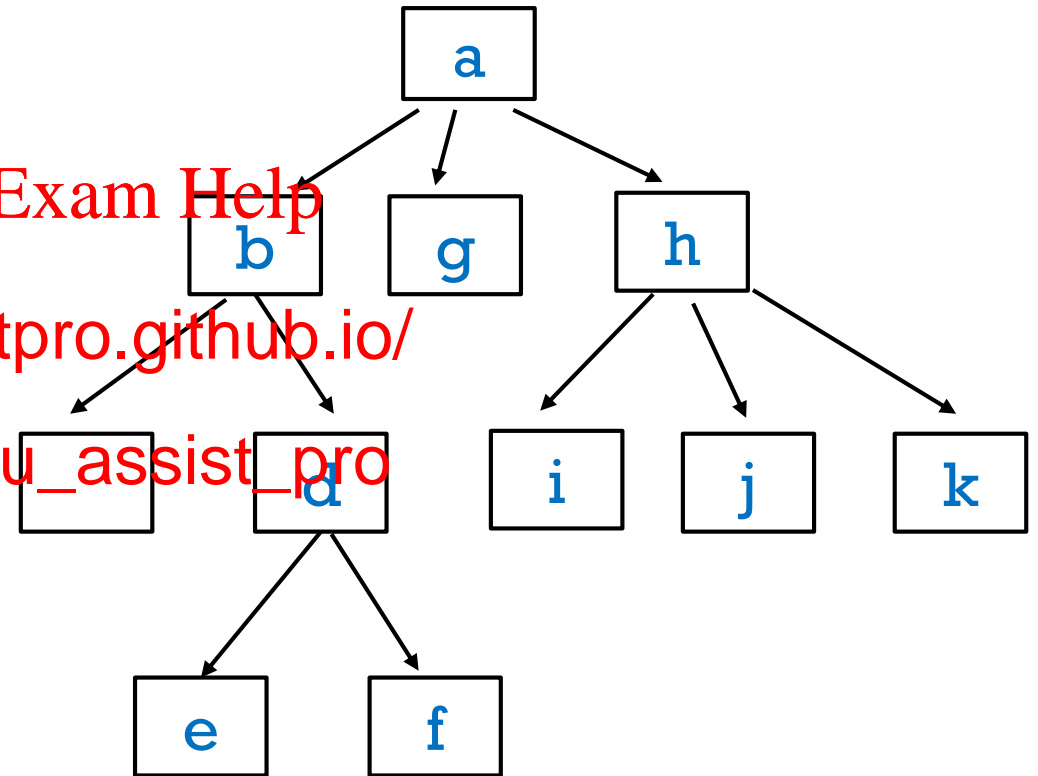
Q: Is it preorder or postorder?

A: It's preorder.

Q: Would move the visit change that?

A: No… why?

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
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 empt
        cur = s.pop()
        visit cur
        for each child of cur
            s.push(child)
    }
}
```

```
treeTraversalUsingQueue(root){
    initialize empty queue q
    ueue(root)
    q is not empty {
        = q.dequeue()
        t cur
        for each child of cur
            q.enqueue(child)
    }
}
```
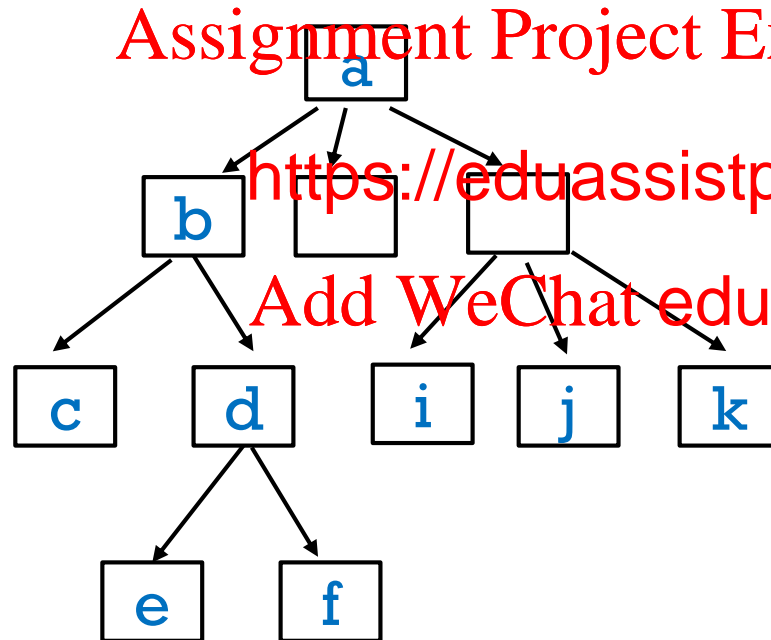
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Queue state at start of the while loop

a

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
treeTraversalUsingQueue(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)
    }
}
```

Queue state at start of the while loop

a

b g h

a

b

c d i j k

e f

```
treeTraversalUsingQueue(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)
        }
}
```
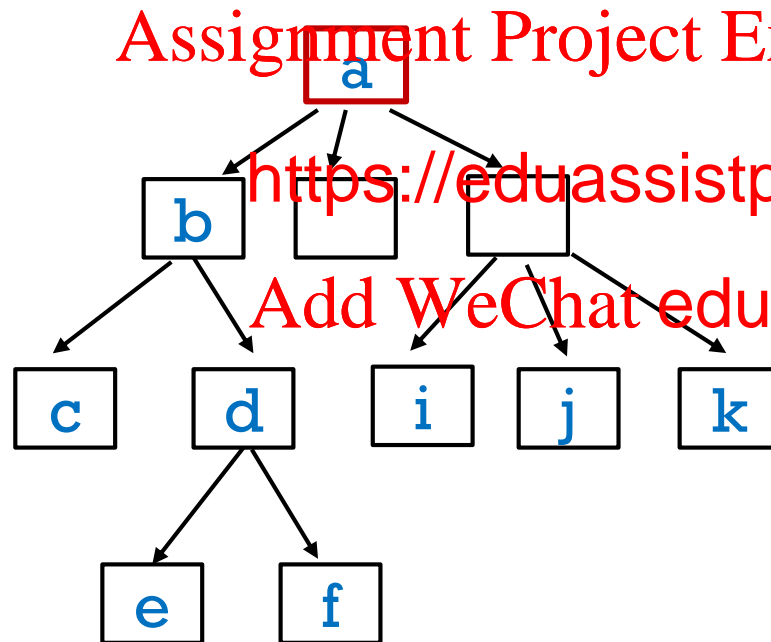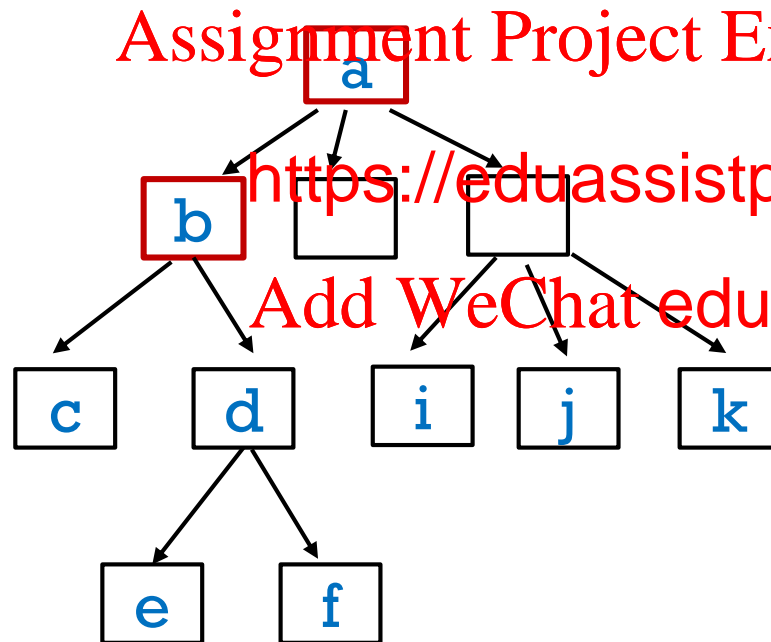
Queue state at start of the while loop

```
a

b g h
g h
g h c d
```

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
treeTraversalUsingQueue(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)
    }
}
```

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://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
                a

        b               
                
    c       d       i   j   k

        e       f
```

```
treeTraversalUsingQueue(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)
    }
}
```
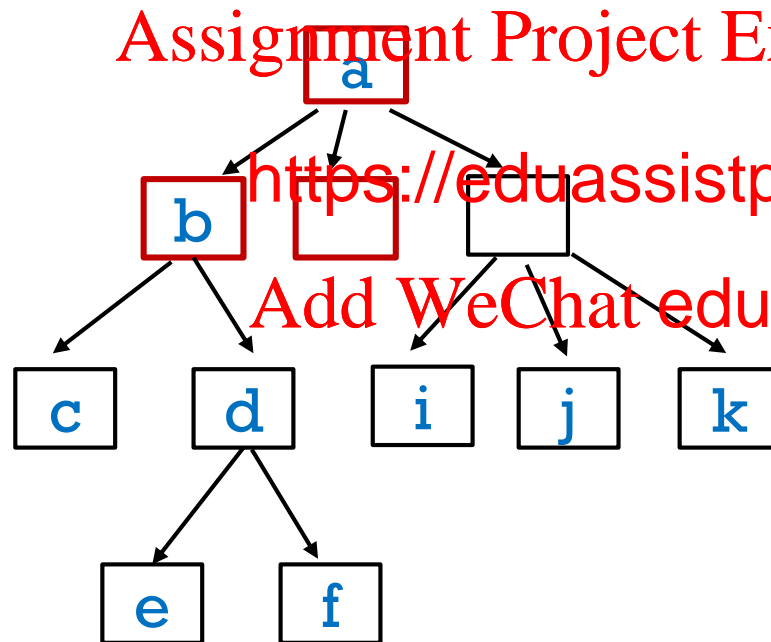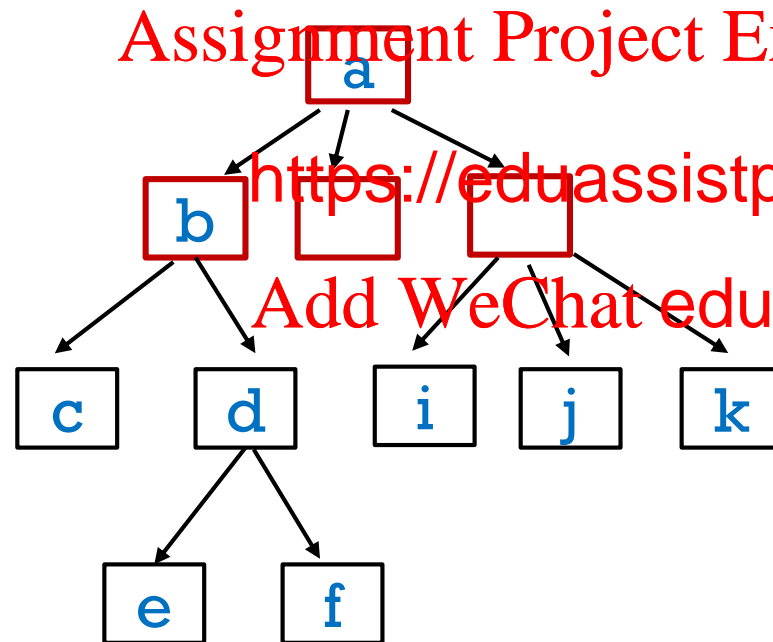
Queue state at start of the while loop

```
a

bgh
gh
ghcd
hcd
cd
cdijk
```

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

```
treeTraversalUsingQueue(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)
    }
}
```

a
b
c d
i j k
e f

Queue state at start of the while loop

a

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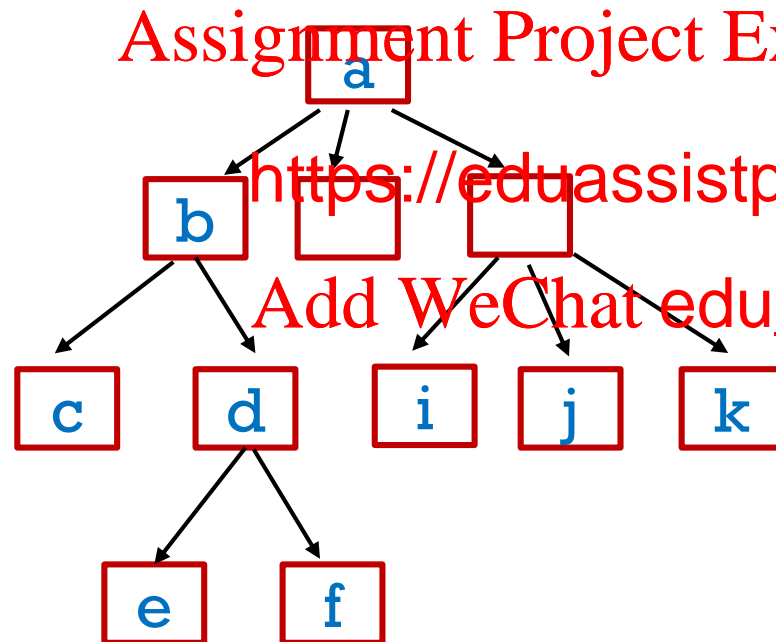g h c d
h c d
c d
c d i j k
d i j k
i j k
i j k e f
j k e f
k e f
e f
f

```
treeTraversalUsingQueue(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)
    }
}
```

a

b

c   d

i   j   k

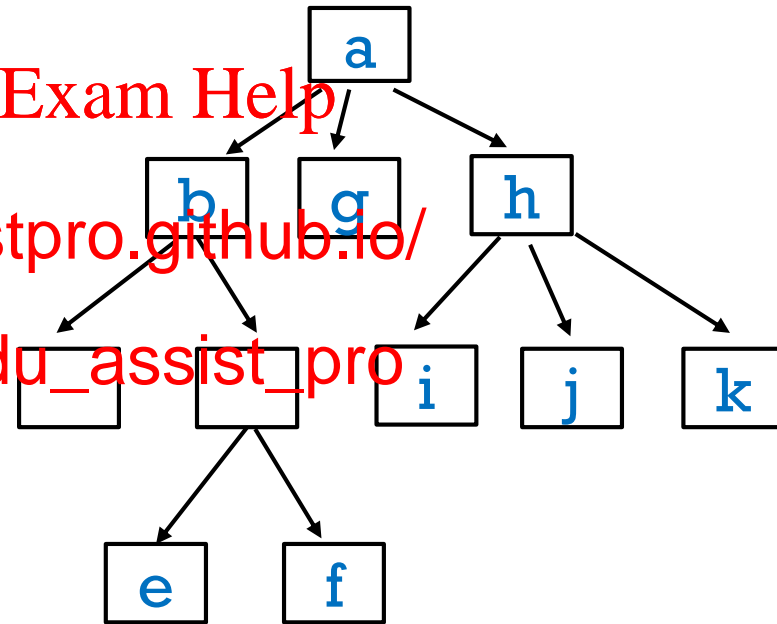e   f

For each level i

   visit all nodes at lev

Order visited: abghcdijkef

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Recall the "first child, next sibling" implementation
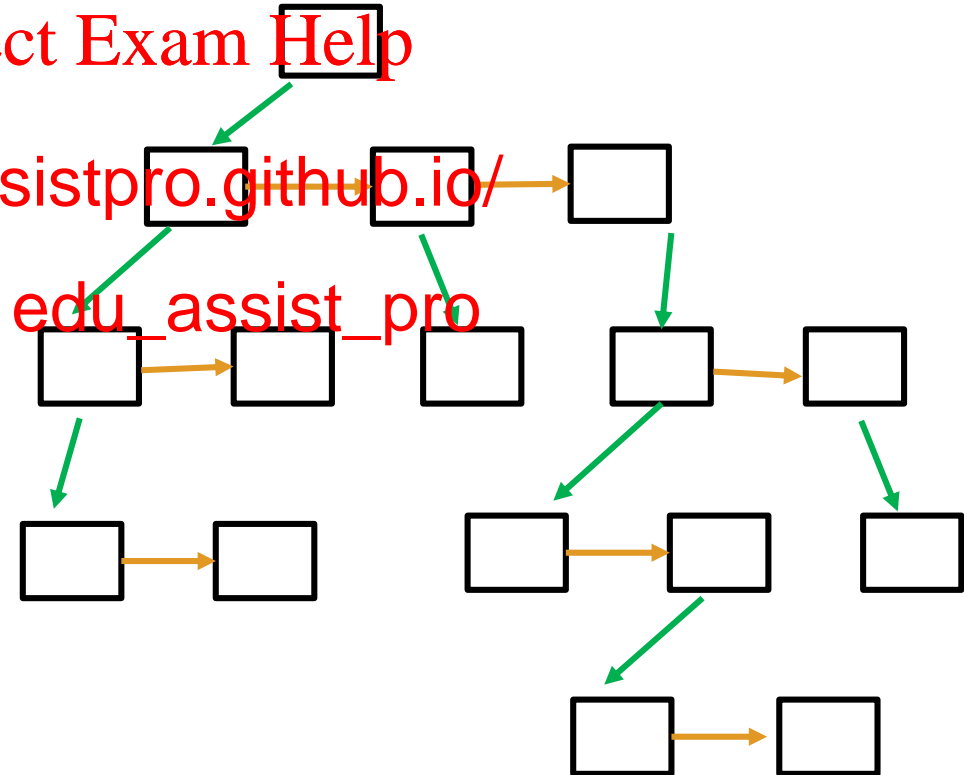
```
class Tree<T>{
   TreeNode<T> root;
        :

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

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

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

}
```
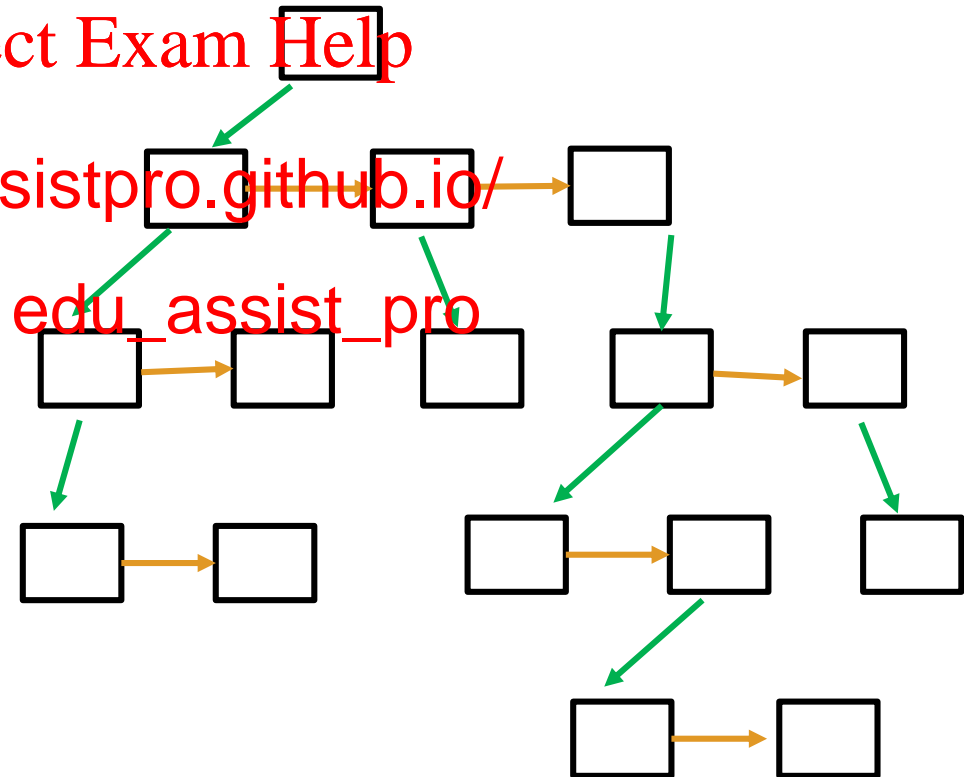
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Coming Soon

Assignment Project Exam Help

In the next

▪ Binary T https://eduassistpro.github.io/

Add WeChat edu_assist_pro