

COMP 250

INTRODUCTION TO COMPUTER SCIENCE

Assignment Project Exam Help

<https://powcoder.com>

Week 11-3: Binary Trees

Giulia Alberini, Fall 2020

Slides adapted from Michael Langer's

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Binary Trees Assignment Project Exam Help

- Expression Trees

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

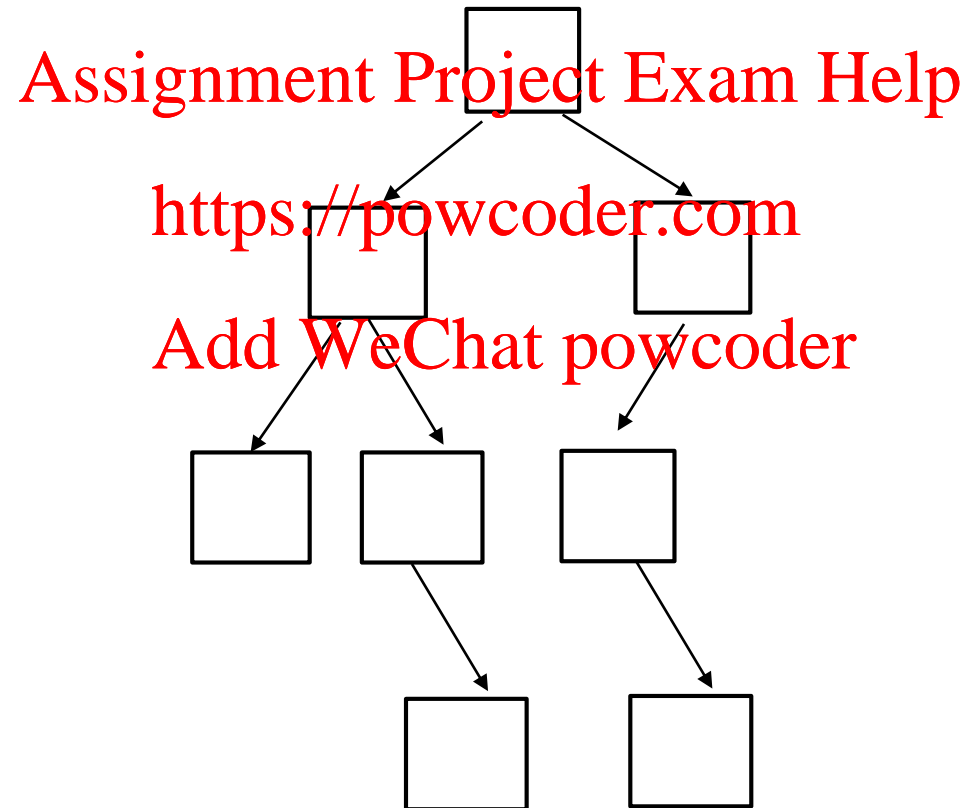
BINARY TREES

<https://powcoder.com>

Add WeChat powcoder

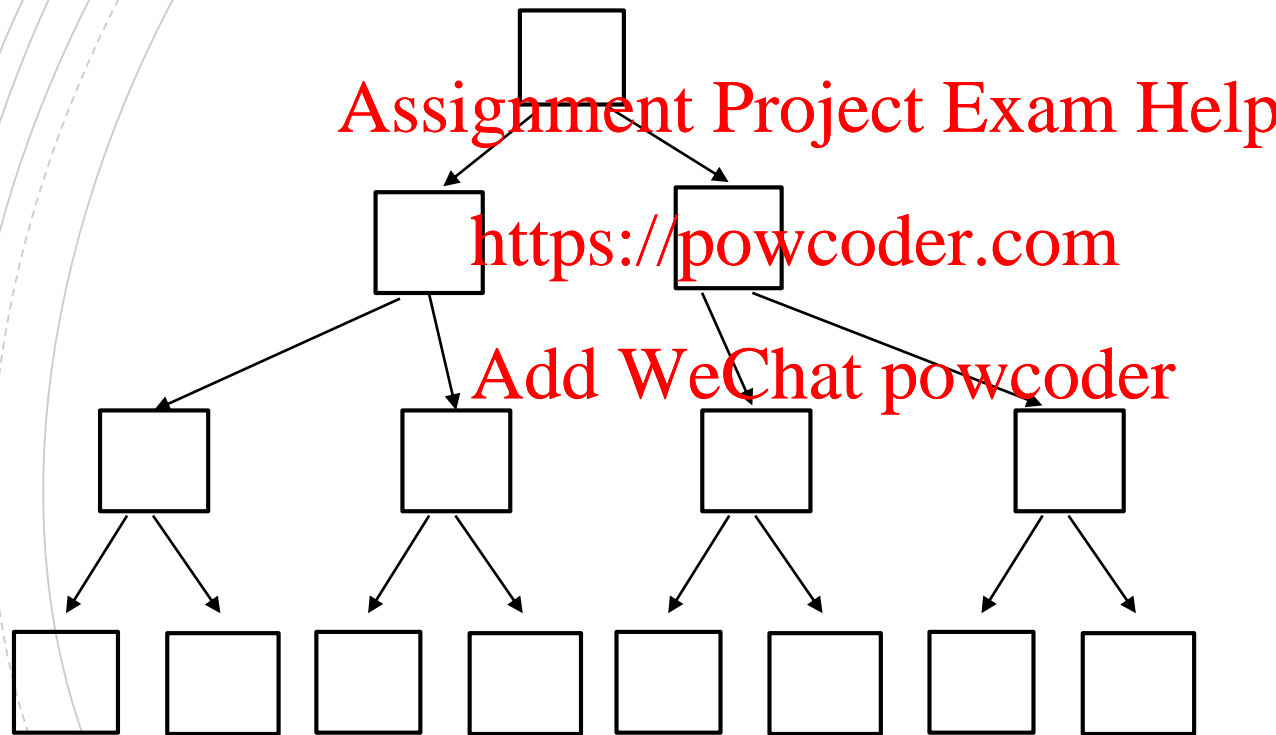
BINARY TREES

Each node has at most 2 children!



BINARY TREES

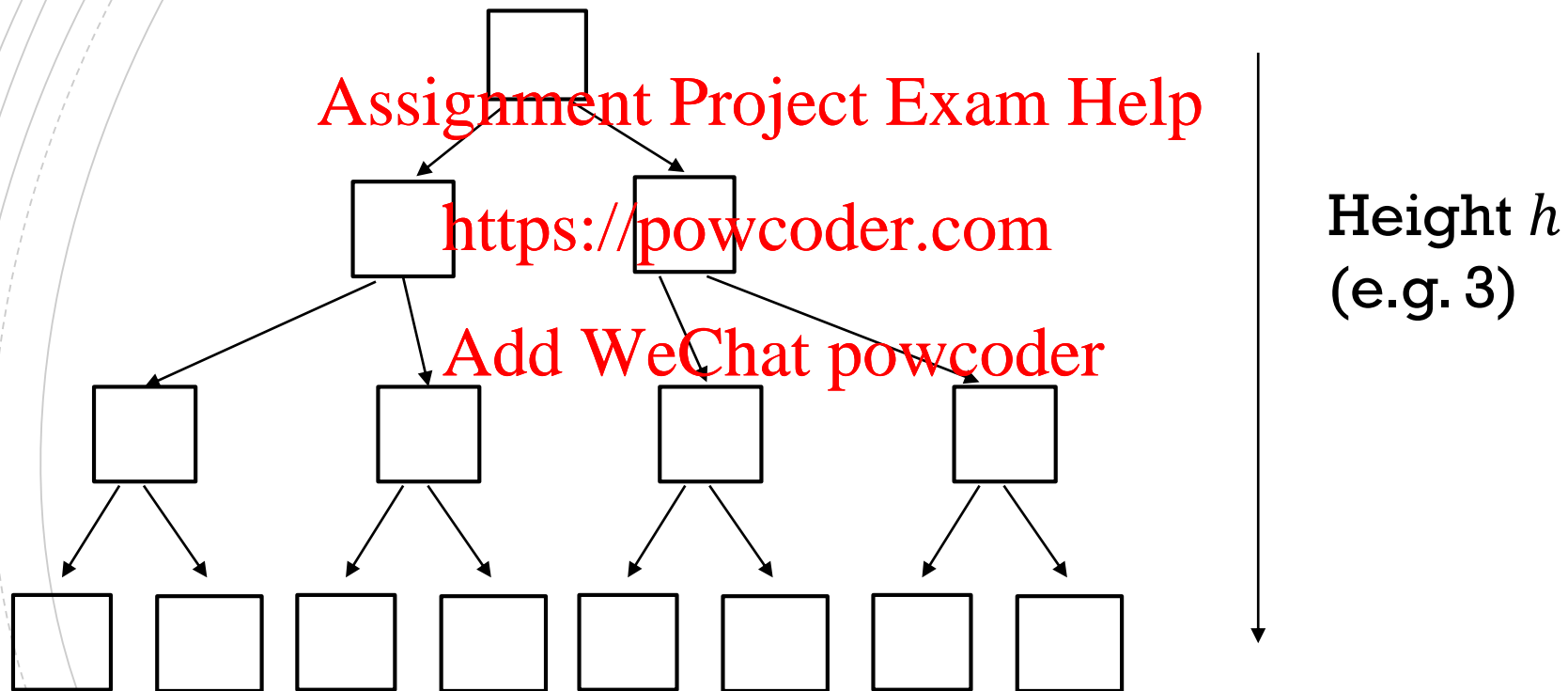
Q: What is the maximum number of nodes n in a binary tree of height h ?



Height h
(e.g. 3)

BINARY TREES

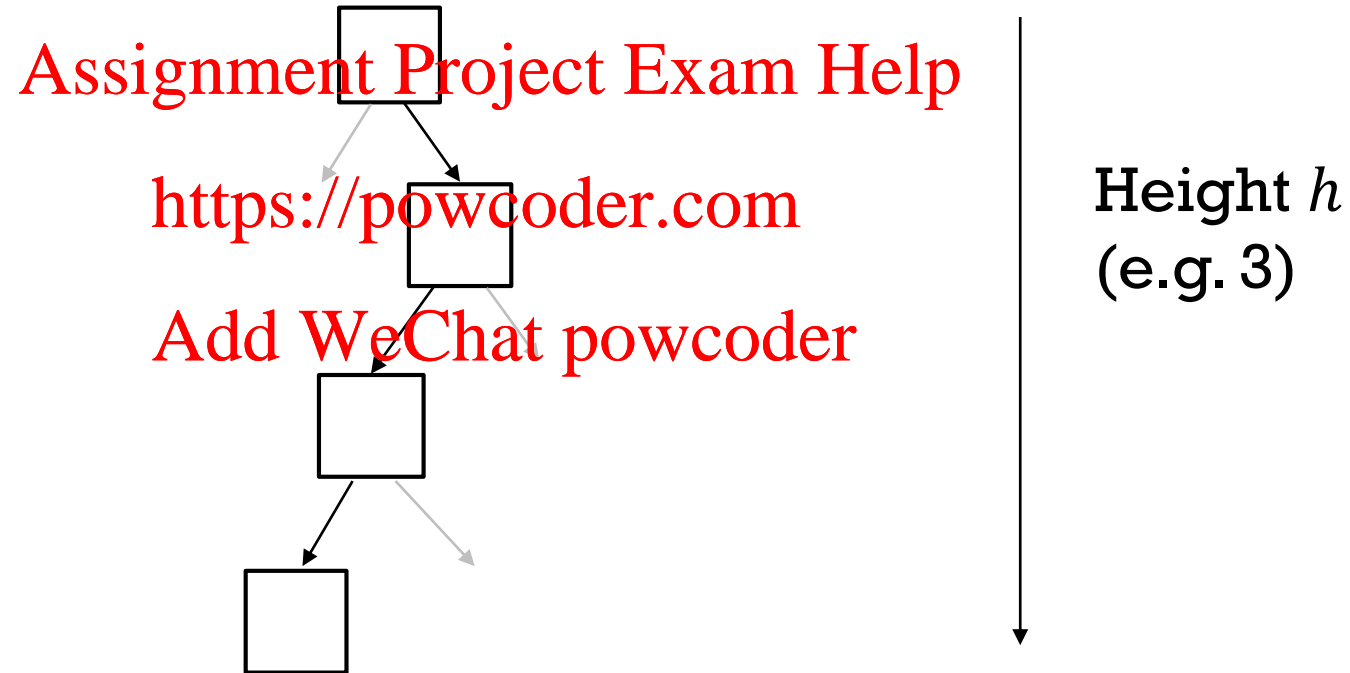
Q: What is the maximum number of nodes n in a binary tree of height h ?



A:
$$n = 1 + 2 + 4 + \dots + 2^h = 2^{h+1} - 1$$

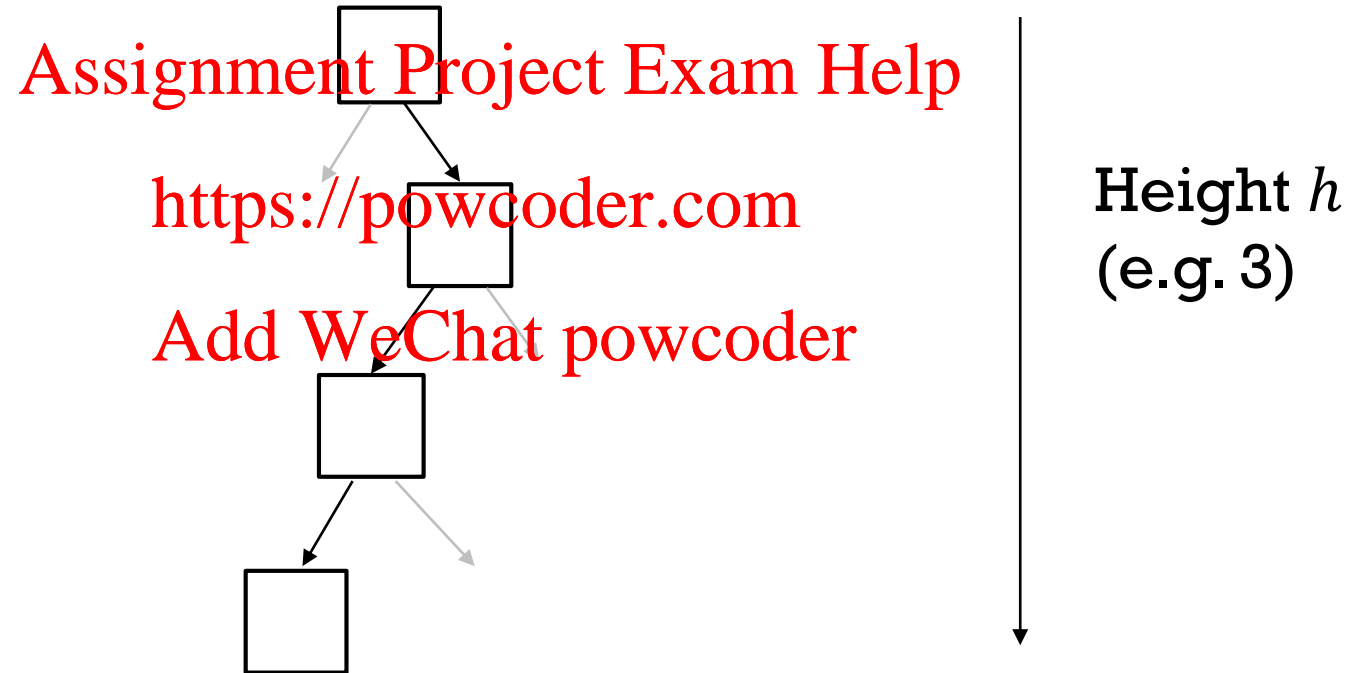
BINARY TREES

Q: What is the minimum number of nodes n in a binary tree of height h ?



BINARY TREES

Q: What is the minimum number of nodes n in a binary tree of height h ?



A: $n = h + 1$

BINARY TREES - IMPLEMENTATION

```
class BTree<T>{
```

```
    BTreeNode<T>    root;
```

Assignment Project Exam Help

<https://powcoder.com>

```
class BTreeNode<T>{
```

Add WeChat powcoder

```
    T element;
```

```
    BTreeNode<T> leftchild;
```

```
    BTreeNode<T> rightchild;
```

```
        :
```

```
    }
```

```
}
```

BINARY TREE TRAVERSAL (DEPTH FIRST)

Rooted Tree
(last lecture)

Binary Tree

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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

BINARY TREE TRAVERSAL (DEPTH FIRST)

Rooted Tree
(last lecture)

Binary Tree

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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

```
preorderBT (root) {  
    if (root is not empty) {  
        visit root  
        preorderBT (root.left)  
        preorderBT (root.right)  
    }  
}
```

BINARY TREE TRAVERSAL (DEPTH FIRST)

```
preorderBT (root) {  
    if (root is not empty) {  
        visit root  
        preorderBT (root.left)  
        preorderBT (root.right)  
    }  
}
```

```
postorderBT (root) {
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

BINARY TREE TRAVERSAL (DEPTH FIRST)

```
preorderBT (root) {  
    if (root is not empty) {  
        visit root  
        preorderBT (root.left)  
        preorderBT (root.right)  
    }  
}
```

```
postorderBT (root) {  
    if (root is not empty) {  
        postorderBT (root.left)  
        postorderBT (root.right)  
        visit root  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

BINARY TREE TRAVERSAL (DEPTH FIRST)

```
preorderBT (root) {
    if (root is not empty) {
        visit root
        preorderBT (root.left)
        preorderBT (root.right)
    }
}
```

```
postorderBT (root) {
    if (root is not empty) {
        postorderBT (root.left)
        postorderBT (root.right)
        visit root
    }
}
```

```
inorderBT (root) {
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

BINARY TREE TRAVERSAL (DEPTH FIRST)

```
preorderBT (root) {  
    if (root is not empty) {  
        visit root  
        preorderBT (root.left)  
        preorderBT (root.right)  
    }  
}
```

```
postorderBT (root) {  
    if (root is not empty) {  
        postorderBT (root.left)  
        postorderBT (root.right)  
        visit root  
    }  
}
```

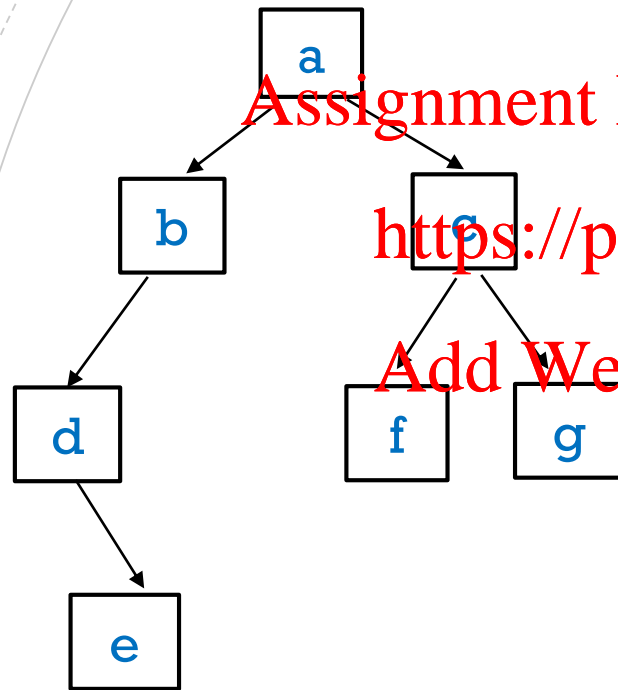
```
inorderBT (root) {  
    if (root is not empty) {  
        inorderBT (root.left)  
        visit root  
        inorderBT (root.right)  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE



Assignment Project Exam Help

Pre order:

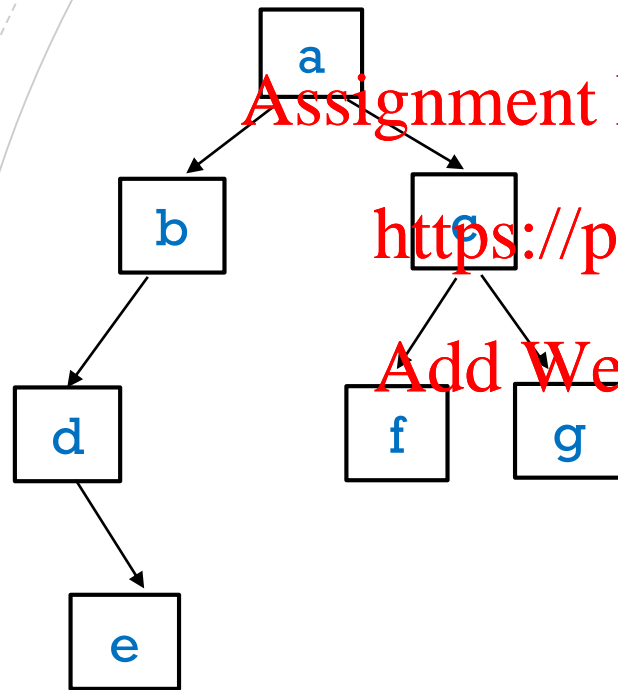
<https://powcoder.com>

In order:

Add WeChat powcoder

Post order:

EXAMPLE



Assignment Project Exam Help

<https://powcoder.com>

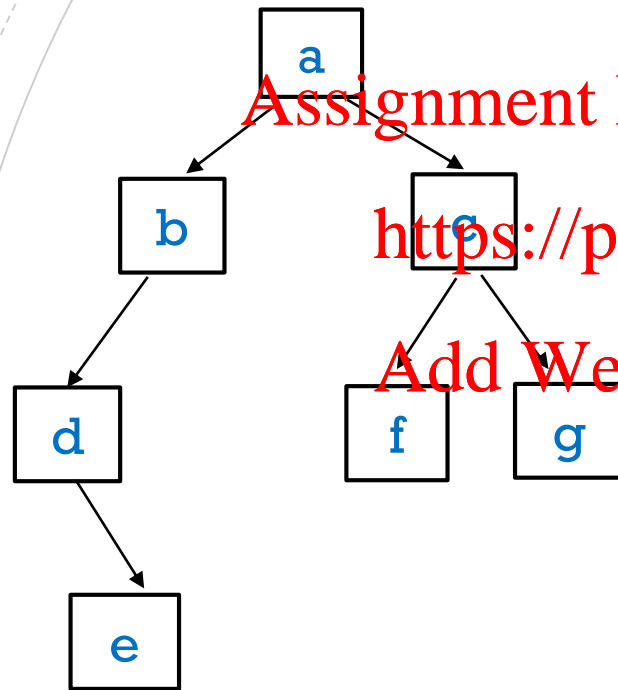
Add WeChat powcoder

Pre order: a b d e c f g

In order:

Post order:

EXAMPLE



Assignment Project Exam Help

<https://powcoder.com>

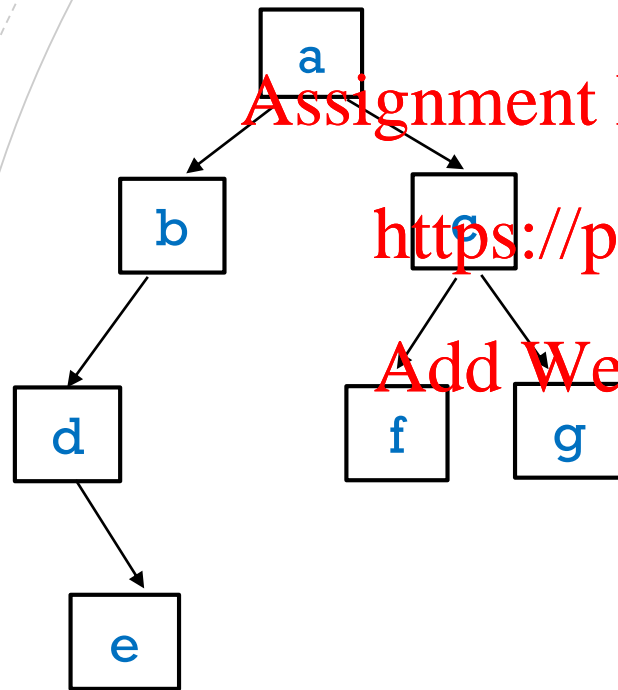
Add WeChat powcoder

Pre order: a b d e c f g

In order: d e b a f c g

Post order:

EXAMPLE



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Pre order: a b d e c f g

In order: d e b a f c g

Post order: e d b f g c a

Assignment Project Exam Help
EXPRESSIONS TREES

<https://powcoder.com>

Add WeChat powcoder

EXPRESSION TREES

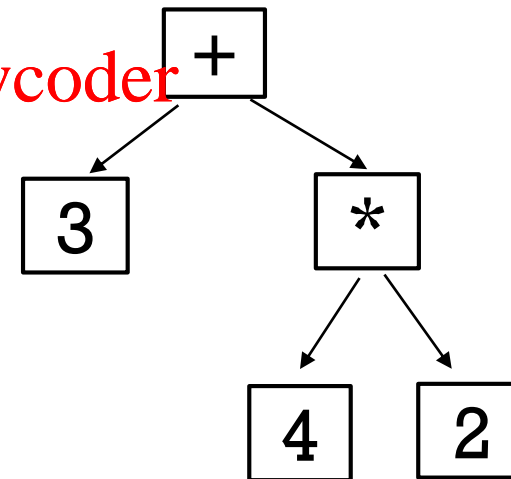
e.g. $3 + 4 * 2$

Assignment Project Exam Help

$3 + (4 * 2)$

<https://powcoder.com>

Add WeChat powcoder



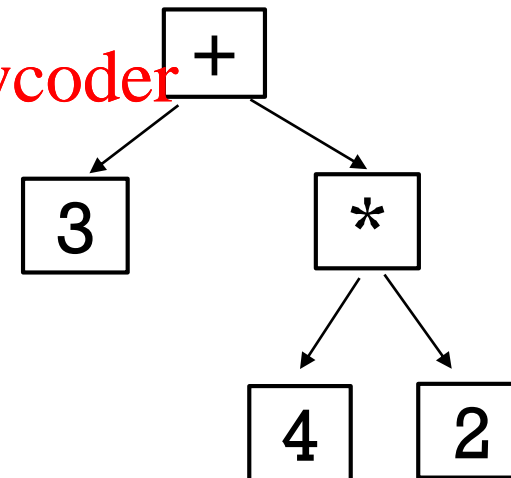
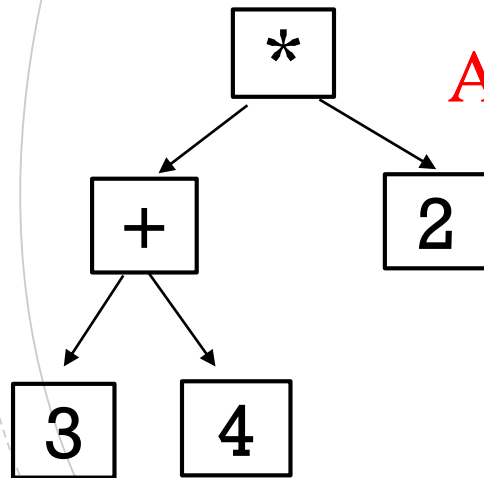
EXPRESSION TREES

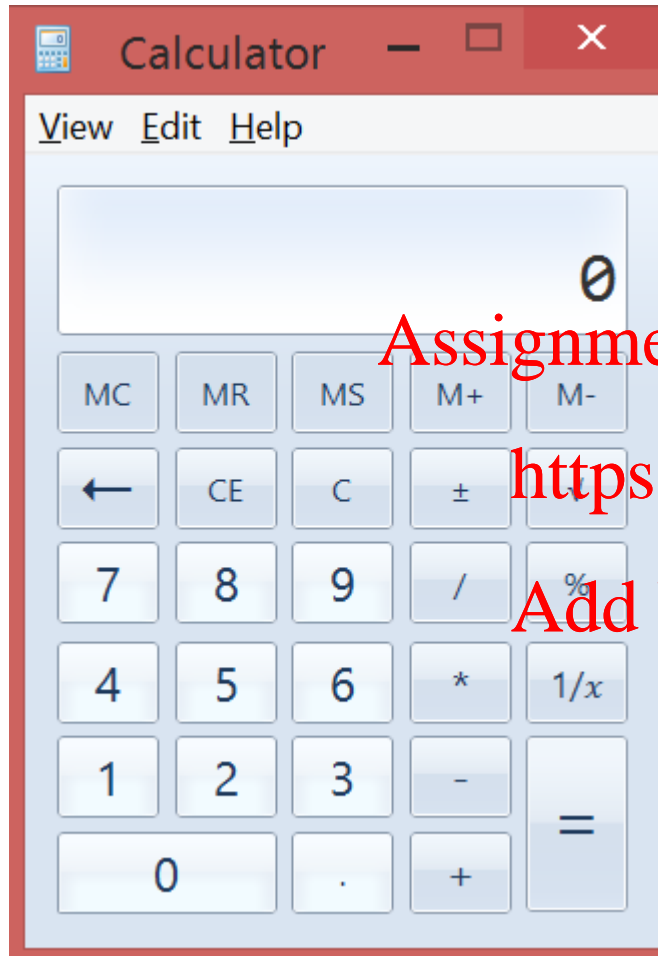
e.g. $3 + 4 * 2$

Assignment Project Exam Help
 $(3 + 4) * 2$ $3 + (4 * 2)$

<https://powcoder.com>

Add WeChat powcoder





My Windows calculator says

$$3 + 4 * 2 = 14.$$

Assignment Project Exam Help

Why? $(3 + 4) * 2 = 14.$

<https://powcoder.com>

Add WeChat powcoder

Whereas....

if I google "3+4*2", I get 11.

$$3 + (4*2) = 11.$$

EXPRESSIONS

We can make expressions using binary operators $+$, $-$, $*$, $/$, $^$

e.g. $a - b / c + d * e ^ f ^ g$

Assignment Project Exam Help

$^$ is exponentiation: $e ^ f ^ g = e ^ (f ^ g)$
<https://powcoder.com>

Add WeChat powcoder

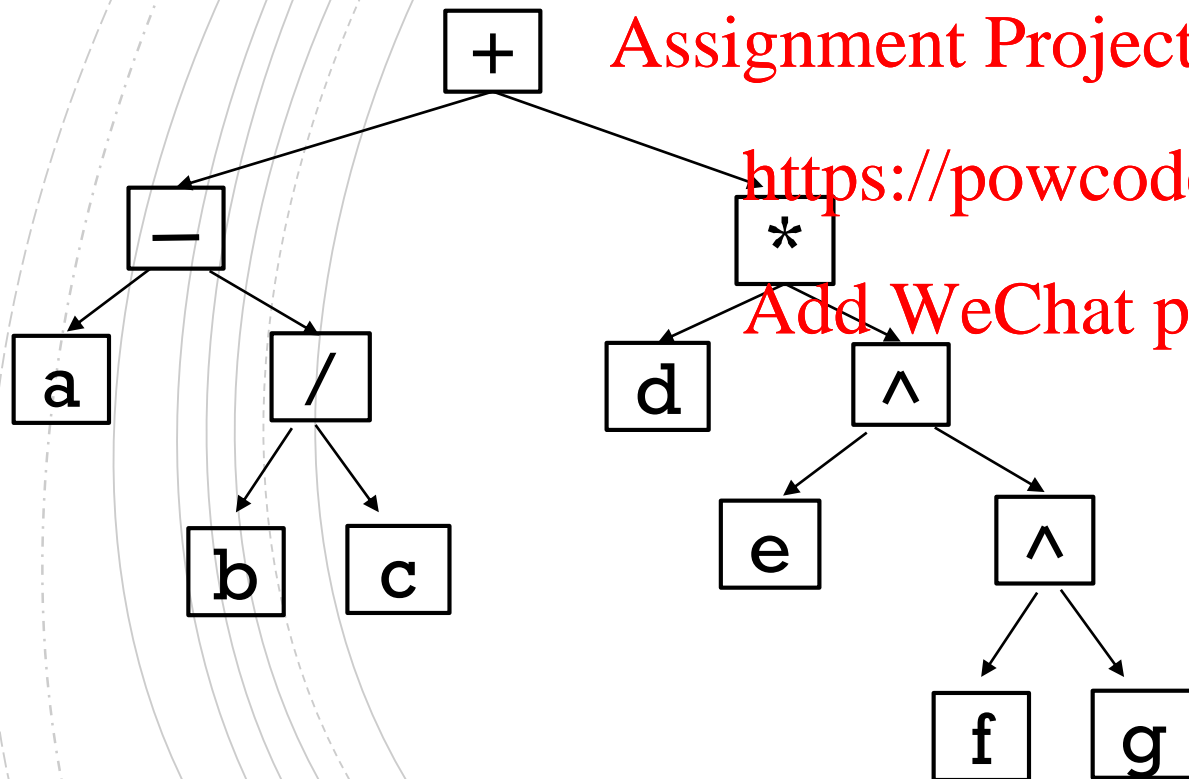
We don't consider unary operators e.g. $3 + -4 = 3 + (-4)$

Operator precedence ordering makes brackets unnecessary.

$$(a - (b / c)) + (d * (e ^ (f ^ g)))$$

EXPRESSION TREE

$$a - b / c + d * e ^ f ^ g \equiv (a - (b / c)) + (d * (e ^ (f ^ g)))$$



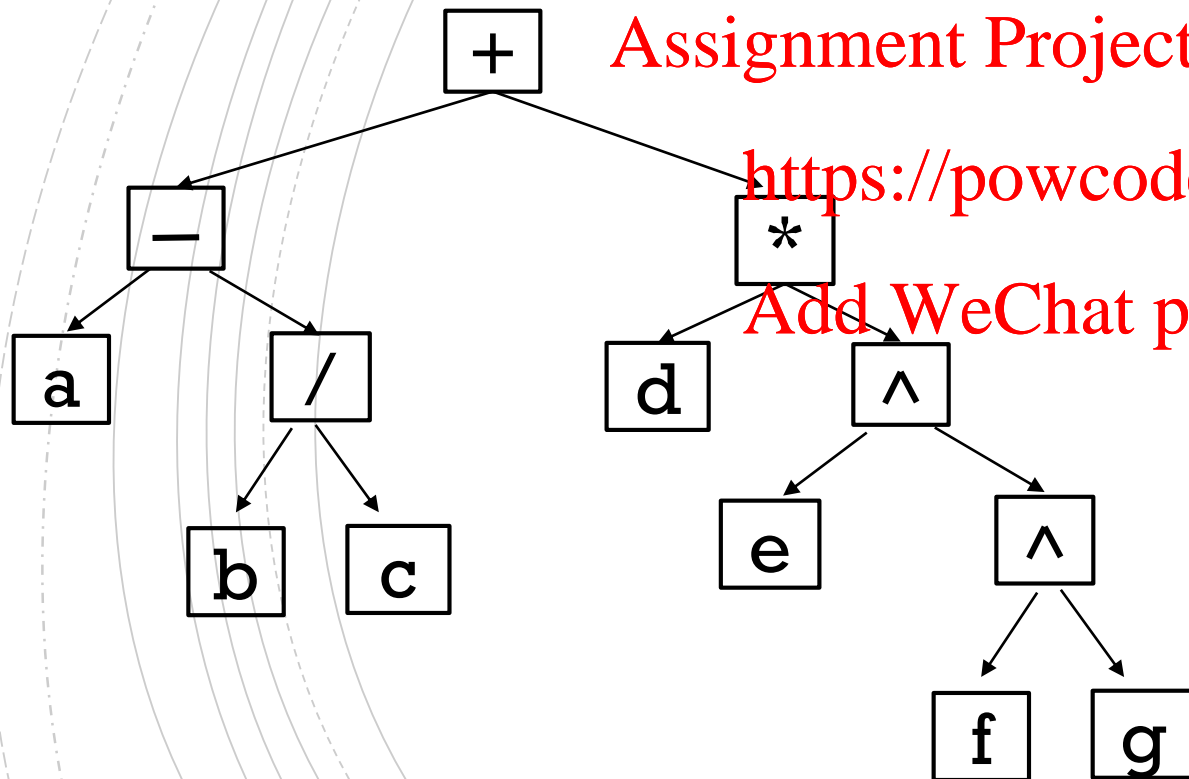
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Internal nodes are *operators*.
Leaves are *operands*.

EXPRESSION TREE



Assignment Project Exam Help

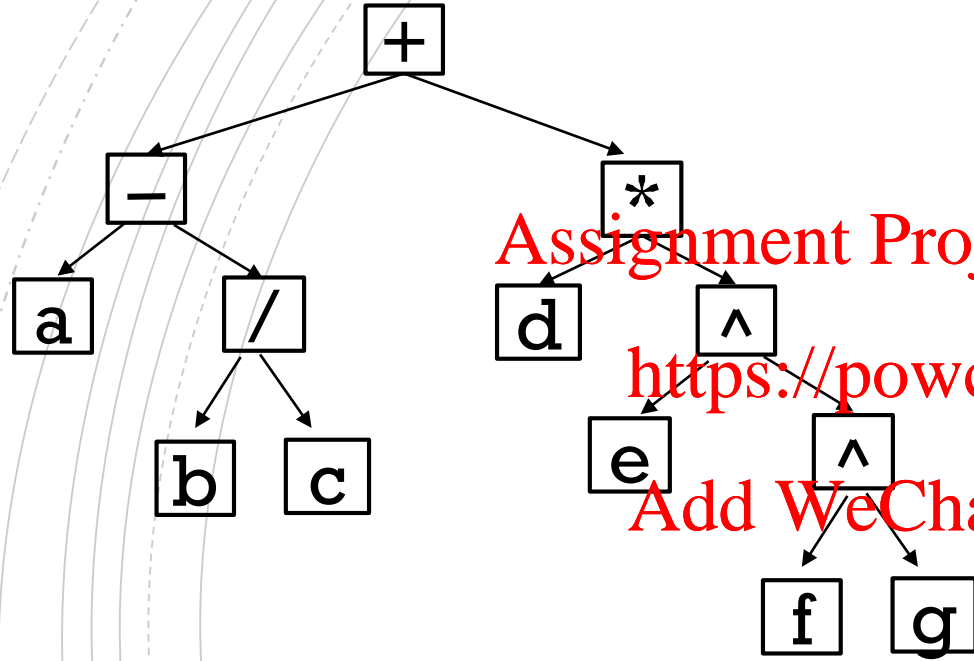
<https://powcoder.com>

Add WeChat powcoder

An expression tree can be a way of *thinking about* the ordering of operations used when evaluating an expression.

But to be concrete, *let's assume we have a binary tree data structure.*

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print out* the node label, what
is the expression printed out?

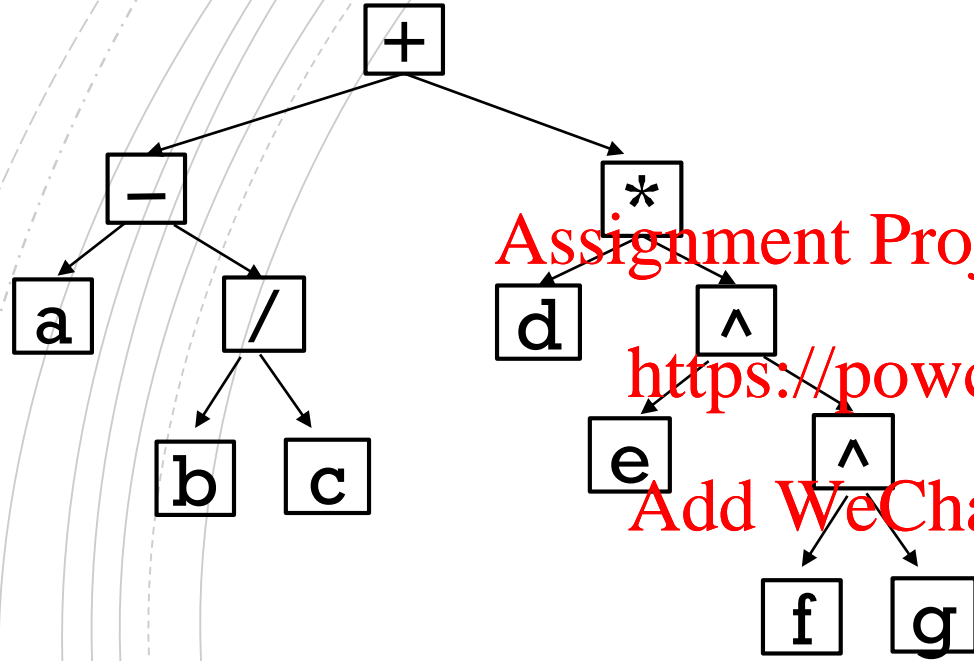
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

preorder traversal gives:

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print* out the node label, what
is the expression printed out?

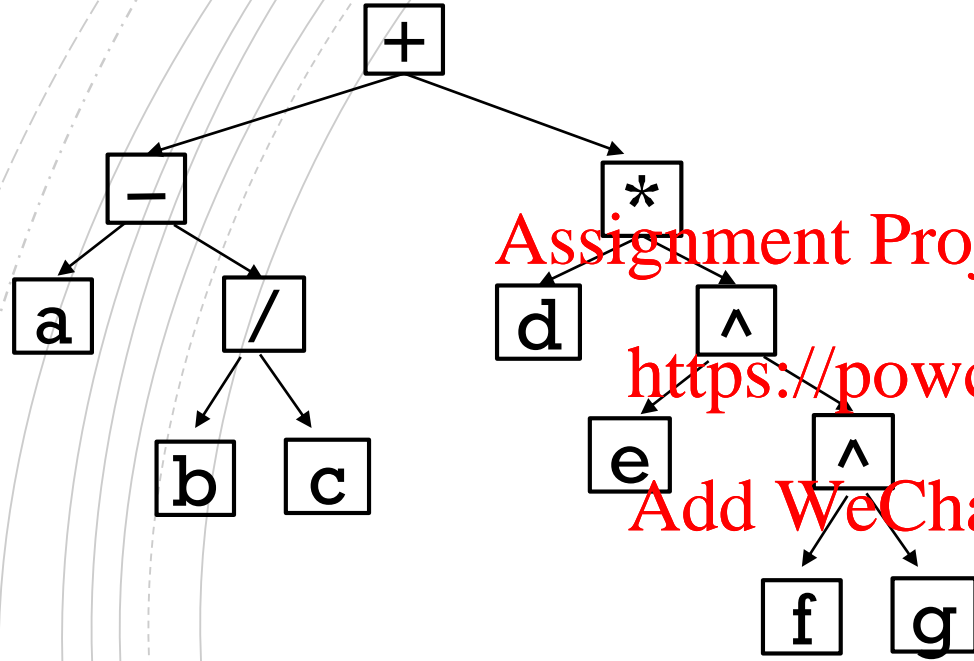
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

preorder traversal gives: $+ - a / b c * d ^ e ^ f g$

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print* out the node label, what
is the expression printed out?

Assignment Project Exam Help

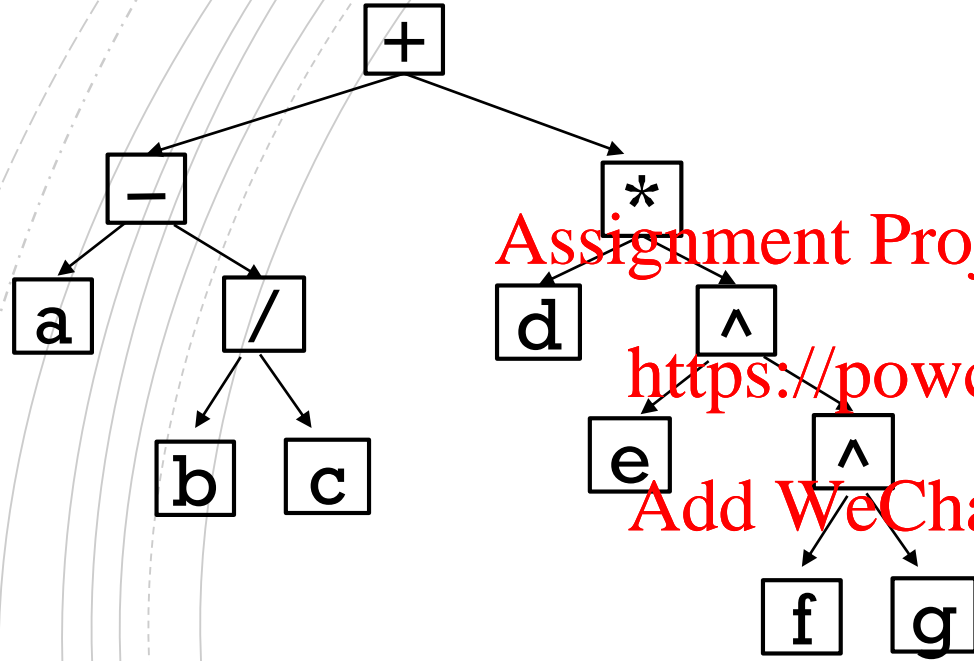
<https://powcoder.com>

Add WeChat powcoder

preorder traversal gives: $+ - a / b c * d ^ e ^ f g$

inorder traversal gives:

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print* out the node label, what
is the expression printed out?

Assignment Project Exam Help

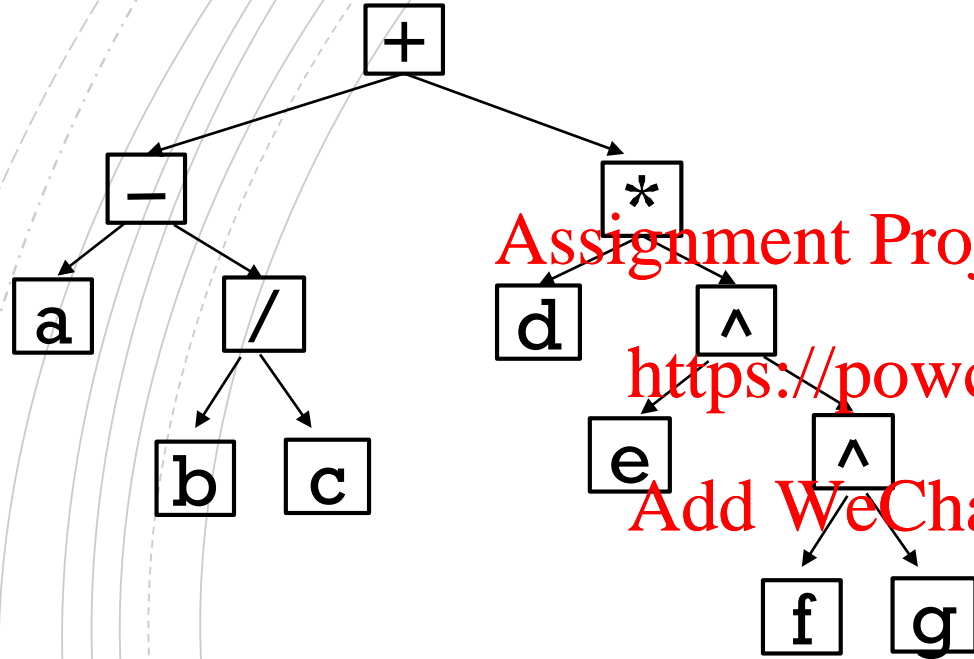
<https://powcoder.com>

Add WeChat powcoder

preorder traversal gives: $+ - a / b c * d ^ e ^ f g$

inorder traversal gives: $a - b / c + d * e ^ f ^ g$

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print* out the node label, what
is the expression printed out?

Assignment Project Exam Help

<https://powcoder.com>

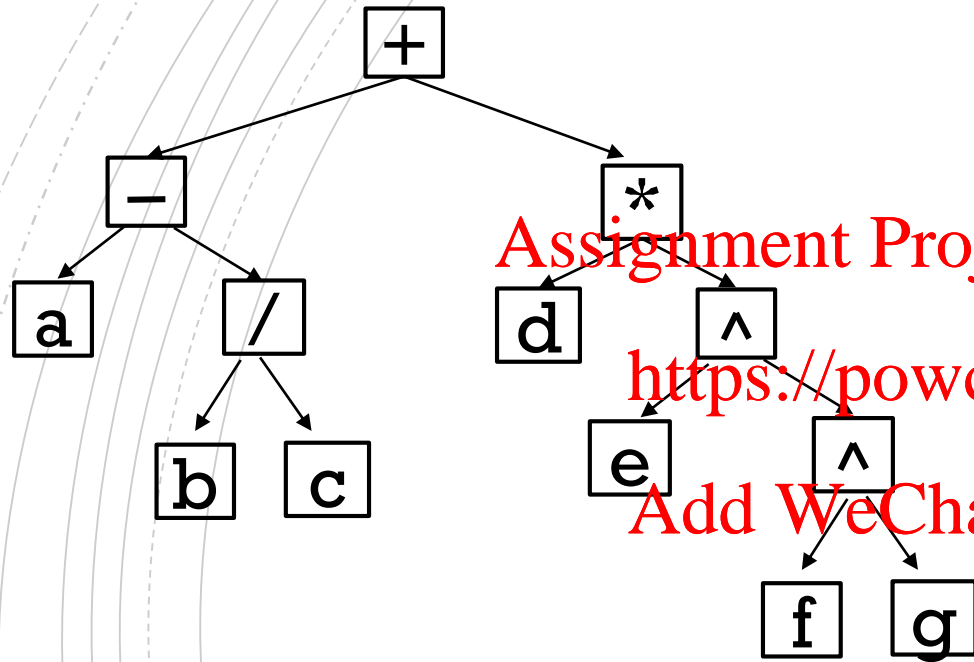
Add WeChat powcoder

preorder traversal gives: $+ - a / b c * d ^ e ^ f g$

inorder traversal gives: $a - b / c + d * e ^ f ^ g$

postorder traversal gives:

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print* out the node label, what
is the expression printed out?

Assignment Project Exam Help

<https://powcoder.com>

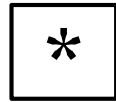
Add WeChat powcoder

preorder traversal gives: $+ - a / b c * d ^ e ^ f g$

inorder traversal gives: $a - b / c + d * e ^ f ^ g$

postorder traversal gives: $a b c / - d e f g ^ ^ * +$

PREFIX, INFIX, POSTFIX EXPRESSIONS



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

prefix: * a b

infix: a * b

postfix: a b *

PREFIX, INFIX, POSTFIX EXPRESSIONS

baseExp = variable | integer

Assignment Project Exam Help

op = + | - | * | / | ^

Add WeChat powcoder

preExp = **baseExp** | **op** **preExp** **preExp**

where | means 'or'

PREFIX, INFIX, POSTFIX EXPRESSIONS

baseExp = variable | integer

Assignment Project Exam Help

op = + | - | * | / | ^

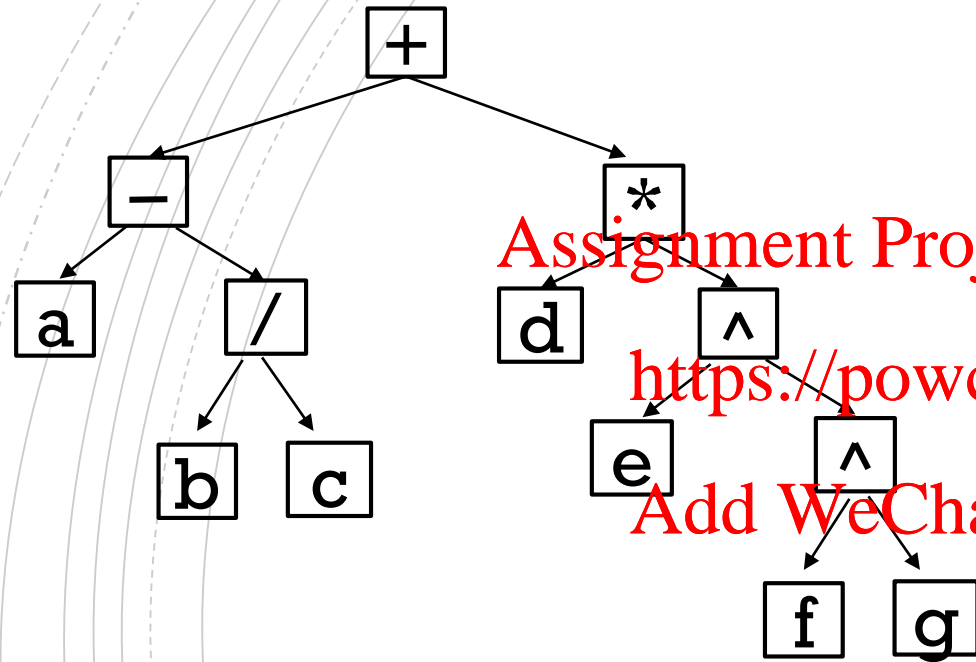
preExp = baseExp | op preExp preExp

inExp = baseExp | inExp op inExp

postExp = baseExp | postExp postExp op

Use
only
one.

EXPRESSION TREE – TRAVERSAL



If we traverse an expression tree,
and *print* out the node label, what
is the expression printed out?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

preorder traversal gives **prefix expression**: $+ - a / b c * d ^ e ^ f g$

inorder traversal gives **infix expression**: $a - b / c + d * e ^ f ^ g$

postorder traversal gives **postfix expression**: $a b c / - d e f g ^ ^ * +$

TERMINOLOGY

- **Prefix** expressions called “Polish Notation”
(after Polish logician Jan Lucasewicz 1920's)
<https://powcoder.com>
Assignment Project Exam Help
- **Postfix** expressions are called “Reverse Polish notation” (RPN)
Add WeChat powcoder

Some calculators (esp. Hewlett Packard) require users to input expressions using RPN.

TERMINOLOGY

- **Prefix** expressions called “Polish Notation”
(after Polish logician Jan Lucasewicz 1920's)

Assignment Project Exam Help

<https://powcoder.com>

- **Postfix** expressions are called “Reverse Polish notation” (RPN)

Add WeChat powcoder

Calculate $5 * 4 + 3$:

5 <enter>

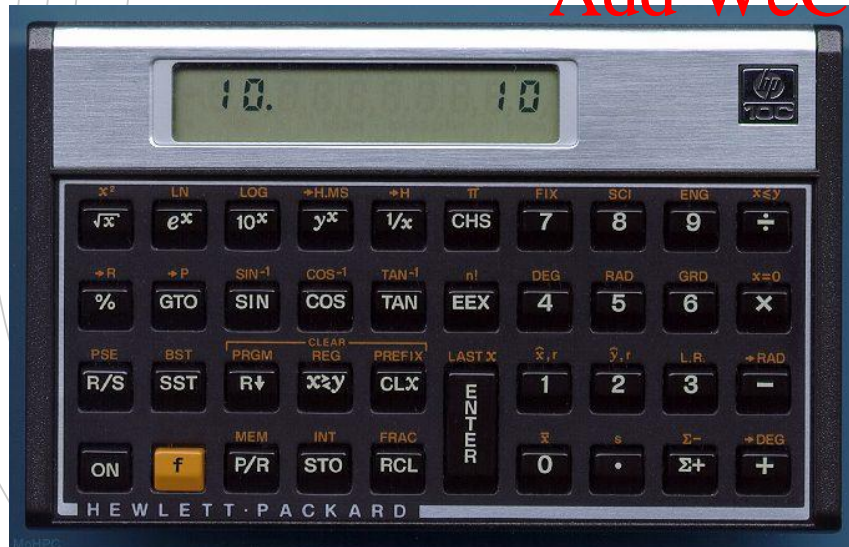
4 <enter>

* <enter>

3 <enter>

+ <enter>

No “=” symbol on keyboard.





Coming Soon

Assignment Project Exam Help

In the next videos:

- Binary Search Trees
- Heaps

<https://powcoder.com>

Add WeChat powcoder