Column

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
    </style>
</head>
<body>
   <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
   </div>
</body>
</html>
```

Given a binary tree, the task is to find the maximum height of the tree. The height of the tree is the number of edges in the tree from the root to the deepest node. For a tree with just one node, the root node, the height of a binary tree is defined to be zero; if there are 2 levels of nodes, the height is 1 and so on. Binary search tree is built according to the usual rules with the following six keys, inserted one at a time given: B, I, N, A, R, Y.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
    </style>
</head>
<body>
    <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
   </div>
</body>
</html>
```

Given a	height of the	edges in the
binary tree,	tree. The	tree from the
the task is to	height of the	root to the
find the	tree is the	deepest node.
maximum	number of	For a tree

with just one defined to be node, the root zero; if there are 2 levels node, the height of a of nodes, the binary tree is height is 1

and so on. Binary search tree is built according to the usual

rules with the given: B, I, following six keys, inserted one at a time

N, A, R, Y.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTE-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
            overflow: auto;
    </style>
</head>
<body>
    <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
    </div>
</body>
</html>
```

Given a maximum tree is the binary tree, height of the number of the task is to tree. The edges in the find the height of the tree from the

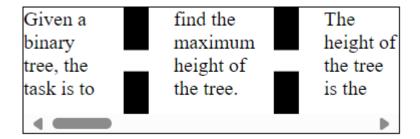
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTE-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
            overflow: auto;
            -webkit-column-gap: 3px;
    </style>
</head>
<body>
    <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
    </div>
</body>
</html>
```

Given a binary height of the number of tree, the task is tree. The edges in the to find the height of the maximum tree is the root to the

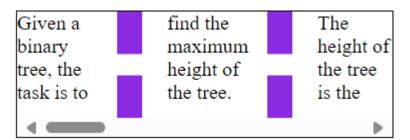
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
            overflow: auto;
            -webkit-column-gap: 3px;
            -webkit-column-rule-style: dashed;
    </style>
</head>
<body>
    <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
    </div>
</body>
</html>
```

Given a binary height of the tree, the task is tree. The to find the maximum tree is the maximum tree is the number of edges in the tree from the root to the

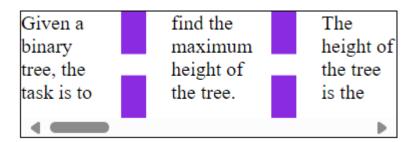
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTE-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
   <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
            overflow: auto;
            -webkit-column-gap: 60px;
            -webkit-column-rule-style: dashed;
            -webkit-column-rule-width: 20px;
    </style>
</head>
<body>
    <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
    </div>
</body>
</html>
```



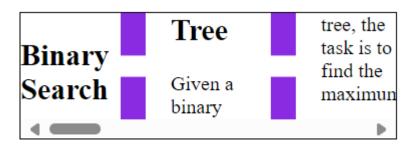
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
            overflow: auto;
            -webkit-column-gap: 60px;
            -webkit-column-rule-style: dashed;
            -webkit-column-rule-width: 20px;
            -webkit-column-rule-color: ■blueviolet;
    </style>
</head>
<body>
    <div id="d1">
        Given a binary tree, the task is to find the maximum height of the
    </div>
</body>
</html>
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <style>
       #d1{
           width: 300px;
           height: 100px;
            border: 1px solid ■ black;
           -webkit-column-count: 3;
           overflow: auto;
            -webkit-column-gap: 60px;
            -webkit-column-rule: dashed 20px ■blueviolet;
   </style>
</head>
<body>
   <div id="d1">
       Given a binary tree, the task is to find the maximum height of the
   </div>
</body>
</html>
```



```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initia</pre>
    <title>Document</title>
    <style>
        #d1{
            width: 300px;
            height: 100px;
            border: 1px solid ■ black;
            -webkit-column-count: 3;
            overflow: auto;
            -webkit-column-gap: 60px;
            -webkit-column-rule: dashed 20px ■blueviolet;
    </style>
</head>
<body>
    <div id="d1">
        <h2> Binary Search Tree</h2>
        Given a binary tree, the task is to find the maximum
    </div>
</body>
</html>
```



Binary Search Tree Given a node. on. Binary binary For a tree, the tree with search task is to tree is just one find the built node, the maximum according root height of to the node, the the tree. height of usual The a binary rules height of with the tree is the tree defined following is the to be six keys, number zero; if inserted of edges there are one at a in the 2 levels time tree from of nodes, given: B, the root the I, N, A, to the height is R, Y. deepest 1 and so

```
#d1 width: 300px;
height: 500px;
border: 1px solid ■ black;
-webkit-column-count: 3;
overflow: auto;
-webkit-column-gap: 30px;
-webkit-column-rule: dashed 20px ■ blueviolet;
-webkit-column-width: 40px;
}

h2{
-webkit-column-span: all;
}
```

Binary Search Tree

Given a node. For a Binary search tree binary tree, tree with the task is is built just one to find the node, the according to maximum the usual root node. height of the height rules with the the tree. of a binary The height following tree is of the tree is defined to six keys, the number be zero; if inserted one of edges in there are 2 at a time the tree levels of given: B, I, from the nodes, the N, A, R, Y. root to the height is 1 deepest and so on.