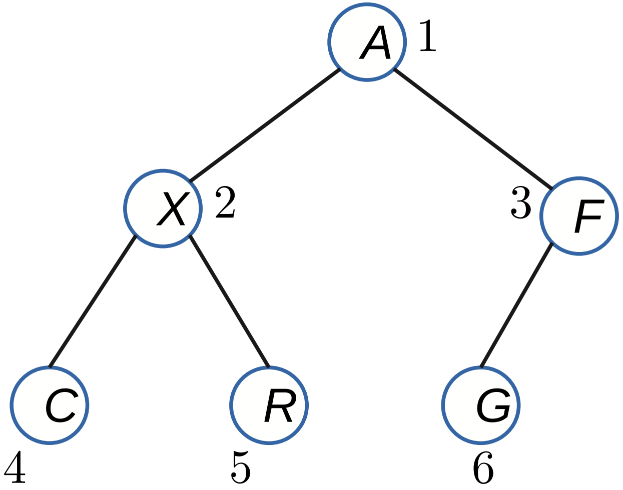
Practice Lessons

Nov. 13 2024

1). Array representation of a binary tree



A
X
F
C
R
G

Input

- The 1st row: an integer n ($1 \leq n \leq 16$), which specifies the number of nodes.
- For the next rows, each row contains three non-negative integers separated by space, say, `x`, `y`, `z`, such that
  - `x` is (the data of) the node,
    `y` is (the data of) its left child, and
    `z` is (the data of) its right child.
  - The 2nd row is designated as the **tree root** node.
- Denote a child by ‘0’ if it does not exist.
- Read the input until `EOF`.

Output: The array representing the binary tree.

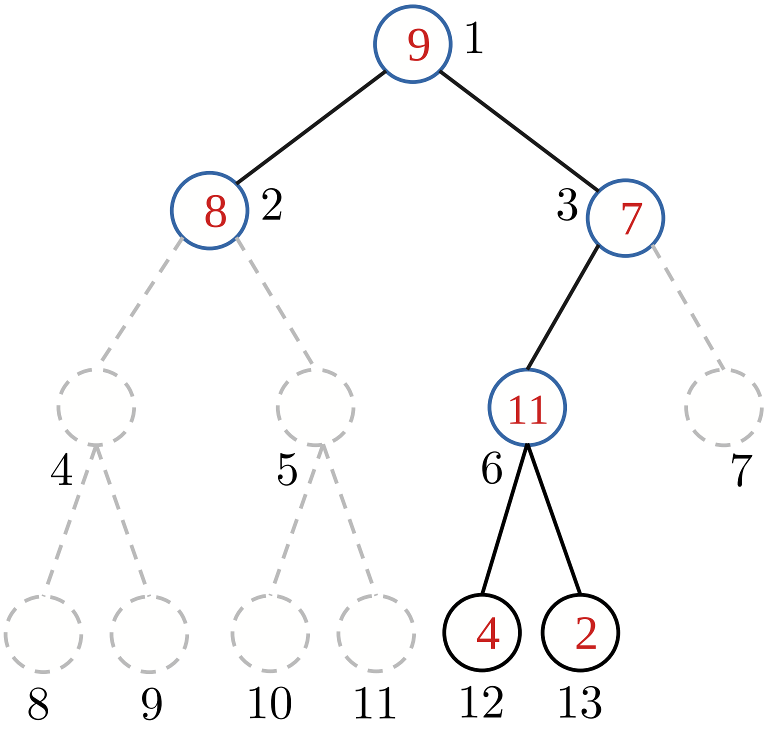
Sample Input & Output

- Sample input:

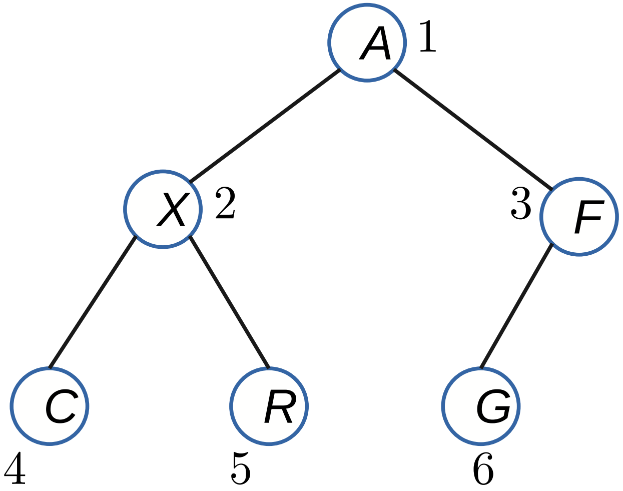
```
6
9 8 7
7 11 0
11 4 2
```

- Sample output:

```
9 8 7 0 0 11 0 0 0 0 0 4 2
```


9
1
8
2
3
7
4
5
11
6
7
4
2
8
9
10
11
12
13

2). Inorder/preorder/postorder traversal



A
X
F
C
R
G

Input

- The 1st row: an integer *n* ($1 \le n \le 16$), which specifies the number of nodes.
- For the next rows, each row contains three non-negative integers separated by space, say, `x`, `y`, `z`, such that
  - `x` is (the data of) the node,
    `y` is (the data of) its left child, and
    `z` is (the data of) its right child.
  - The 2nd row is designated as the **tree root** node.
- Denote a child by ‘0’ if it does not exist.
- Read the input until `EOF`.

Output: The inorder, preorder, and postorder traversals of the binary tree.

Sample Input & Output

- Sample input:

```
6
9 8 7
7 11 0
11 4 2
```

- Sample output:

```
8 9 4 11 2 7
9 8 7 11 4 2
8 4 2 11 7 9
```