Computer Programming 1 Lab

2020-12-24

Chang, Chi-Hung

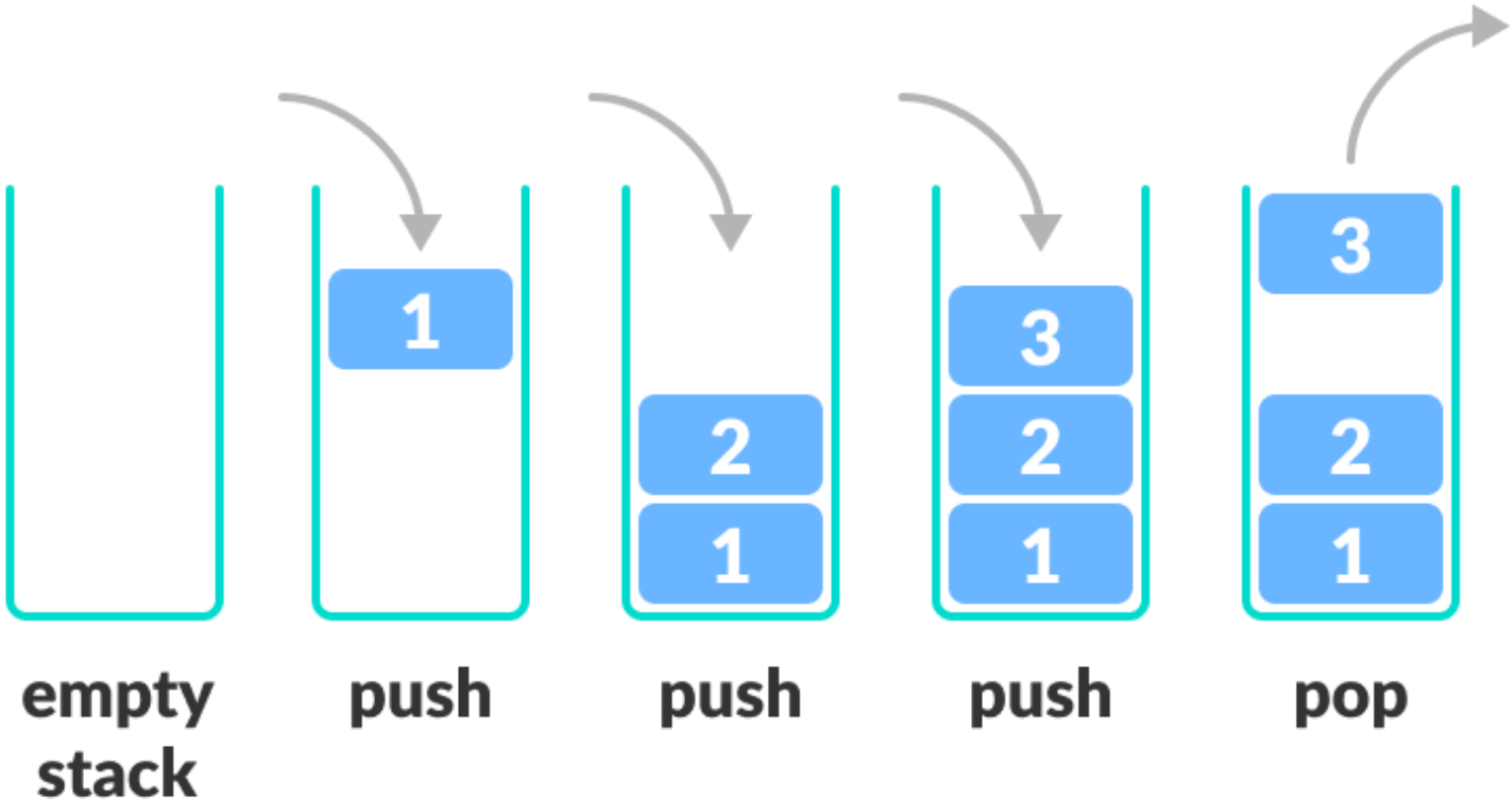
Outline

- Stack
- Queue
- Tree
- Exercise 12

Stack

- Last In, First Out (LIFO)
- Operations
  - push
  - pop

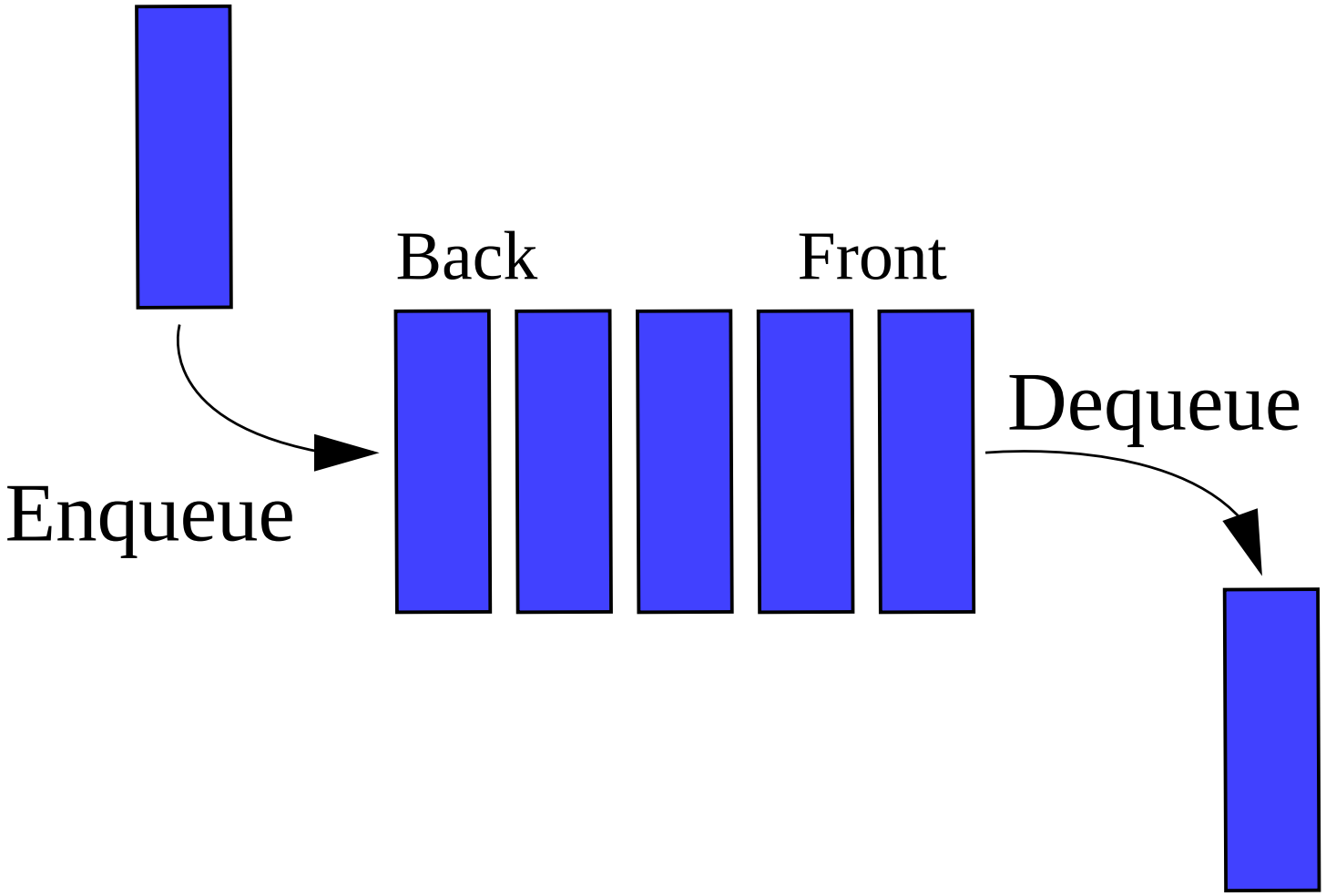
Stack



Queue

- First In, First Out (FIFO)
- Operations
  - enqueue
  - dequeue

Queue



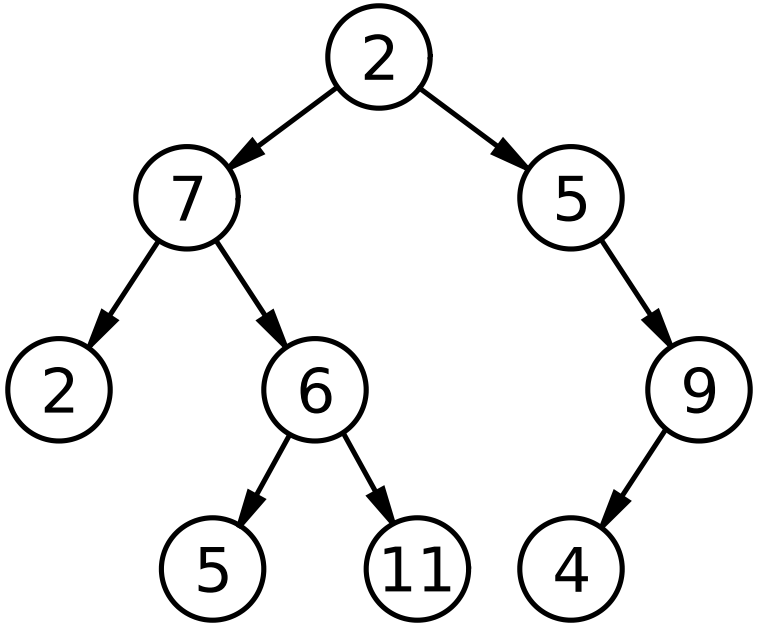
Tree

- Parent & child
- A parent can have more than one child
- All nodes are connected by pointers
- A node with no child is called **leaf**

Tree

Binary Tree

The most common tree that for any parents, there will be at most 2 children



Tree

Tree Structure

```
struct node{
    int value;
    struct node* left;
    struct node* right;
}

struct node* root = malloc(sizeof(struct node));
```

Tree

Insertion

Suppose A is a node that we are going to insert a new node B to its left child.

```
struct node* B = malloc(sizeof(struct node));
B->left = NULL;
B->right = NULL;
A->left = B;
```

Tree

Deletion

Suppose we want to remove A's left node.

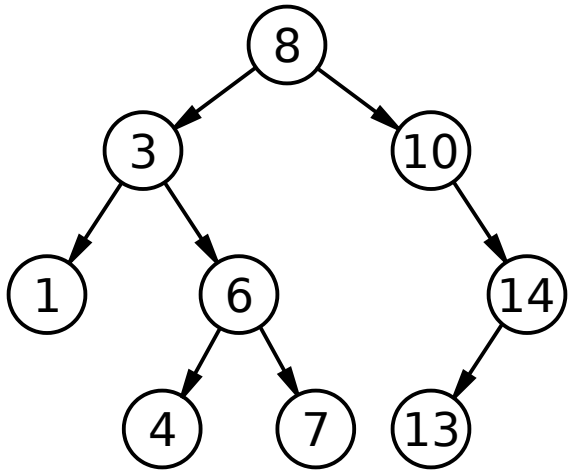
```
free(A->left);
A->left = NULL;
```

Tree

Tree Order

Inorder

Order: **left** $\rightarrow$ **root** $\rightarrow$ **right**



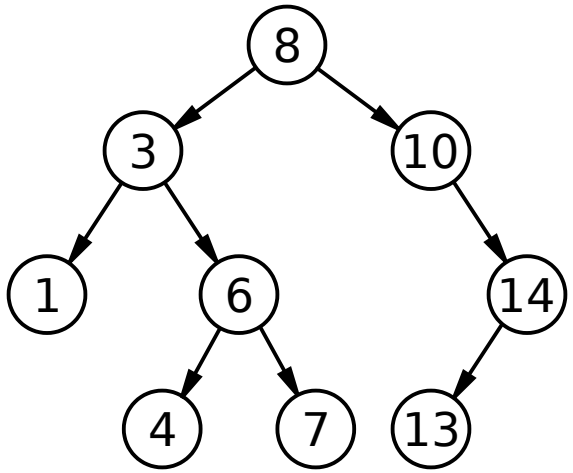
$$1 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 13 \rightarrow 14$$

Tree

Tree Order

Preorder

Order: **root** $\rightarrow$ **left** $\rightarrow$ **right**



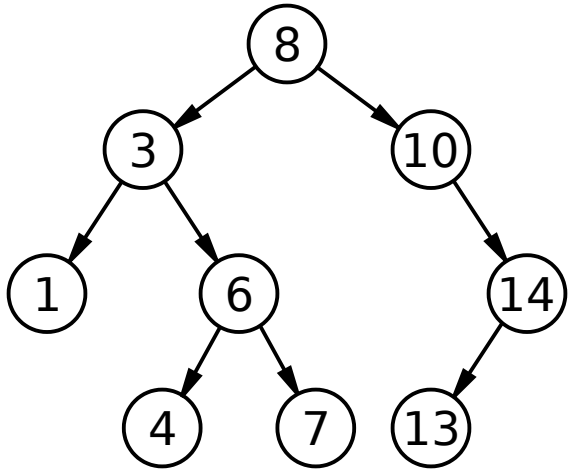
$$8 \rightarrow 3 \rightarrow 1 \rightarrow 6 \rightarrow 4 \rightarrow 7 \rightarrow 10 \rightarrow 14 \rightarrow 13$$

Tree

Tree Order

Postorder

Order: **left** $\rightarrow$ **right** $\rightarrow$ **root**



$$1 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow 13 \rightarrow 14 \rightarrow 10 \rightarrow 8$$

Exercise 12

Give you the elements of a binary tree. The first element is the root node. When getting a new node (number), you should add the node into the binary tree. Please create the binary tree and print out the tree in **postorder**.

- Input

```
8 3 10 1 6 14 4 7 13
```

- Output

```
1 4 7 6 3 13 14 10 8
```

Any Question?