

C++

Stack<int> s

s.push(5)
s.push(4)
s.push(3)
s.pop() → return
s.top() = 4
s.pop()
s

Stack

push()

insert()

pop()

top()

#1

class Stack

{

push → Void

pop → int

top → int

isEmpty() → TRUE
False → bool

};



class Stack

{
stack() {
}

void push(int num) {
//code
}

s.push

#2



implementation



Linked List

Stack
Queue
Heap



predefined

Node

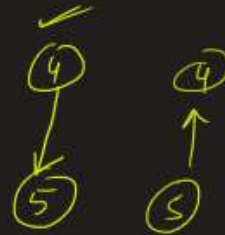
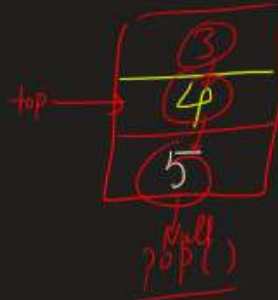
class Stack



class

Node

pop {
top = 5
}



#3

Node* a = new Node(4);
Node* b = new Node(3);
a->next = b

int data;
int *next;

Node(int data, int *next) {
data = data;
}

int *next = NULL; }
By def

a->next = c
a->data = 7



#4

class
Size = 1
head = 4

class Stack {

Node* head = new Node(-1)

int size = 0

Stack() {

}

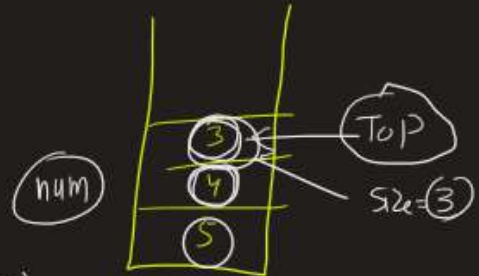
void push(int num) {

if (size == 0) {
Node* a = new Node(num);
head = a;
size++
}

else {

}

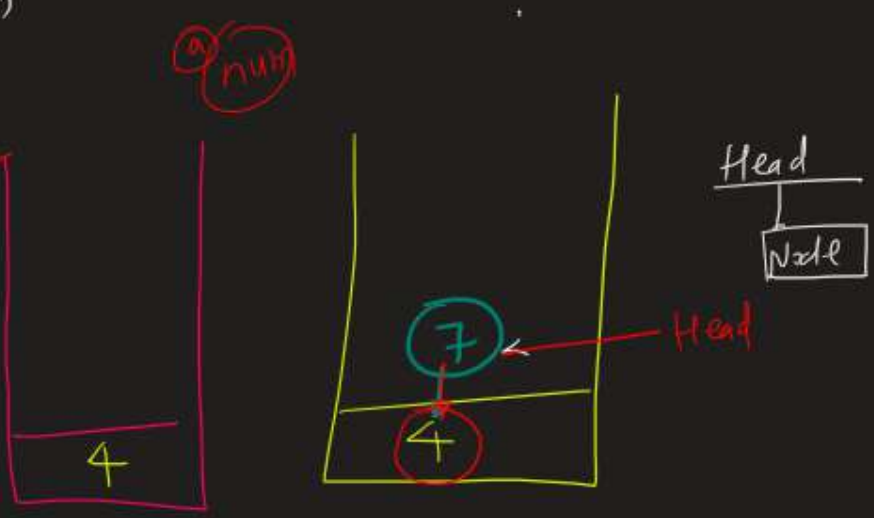
{



Top = Assign

s.push(14)
s.push(7)

$\text{Node * a = newNode(num)}$
 a.next = head
 head = a



Class {
Head
size

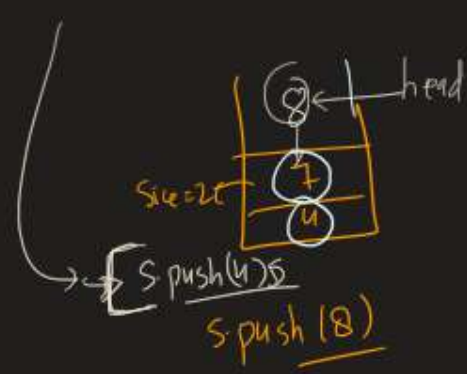
(\$)

Void push (int num)

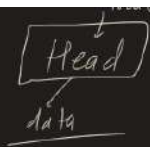
1 Node *a = new Node(num); ← a ○

2 if (size == 0)
3 Head = a
4 size++;

5 else
6 a.next = head
7 head = a;
8 size++;

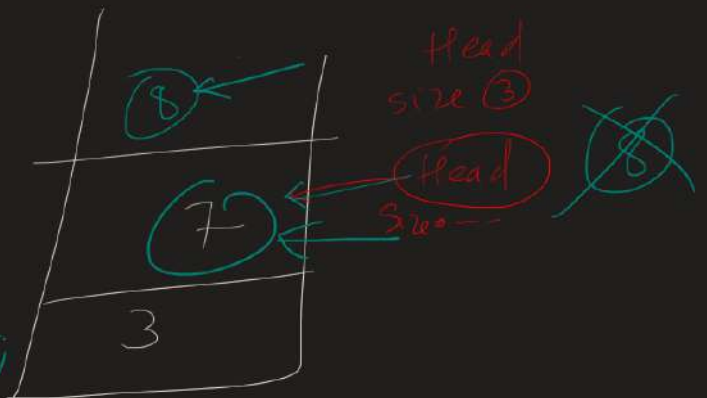


```
int top() {
    return head->data;
}
```

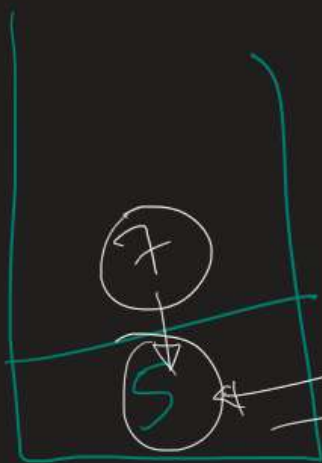


#6

```
int pop() {
    if (size == 0) return -1;
    else {
        size--;
        int value = head->data;
        head = head->next;
    }
}
```



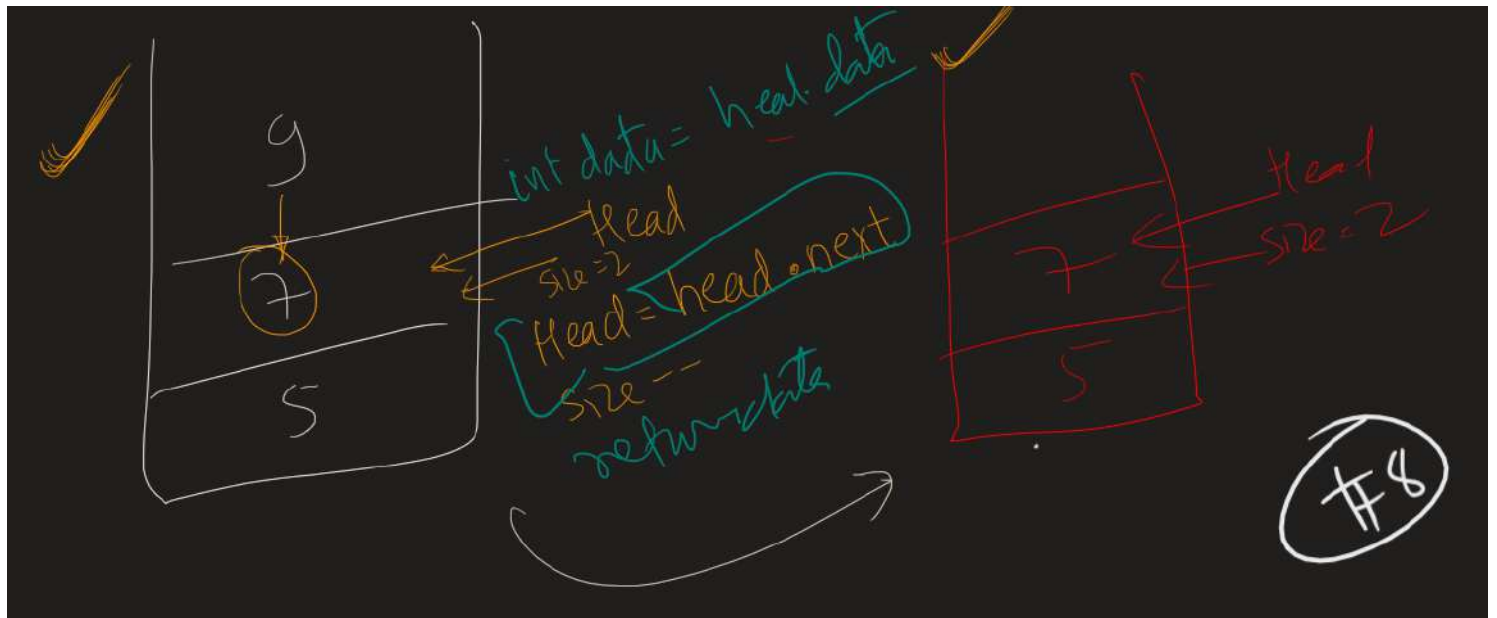
s.push(7)

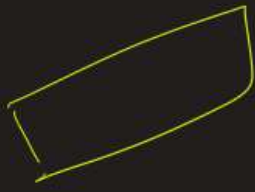
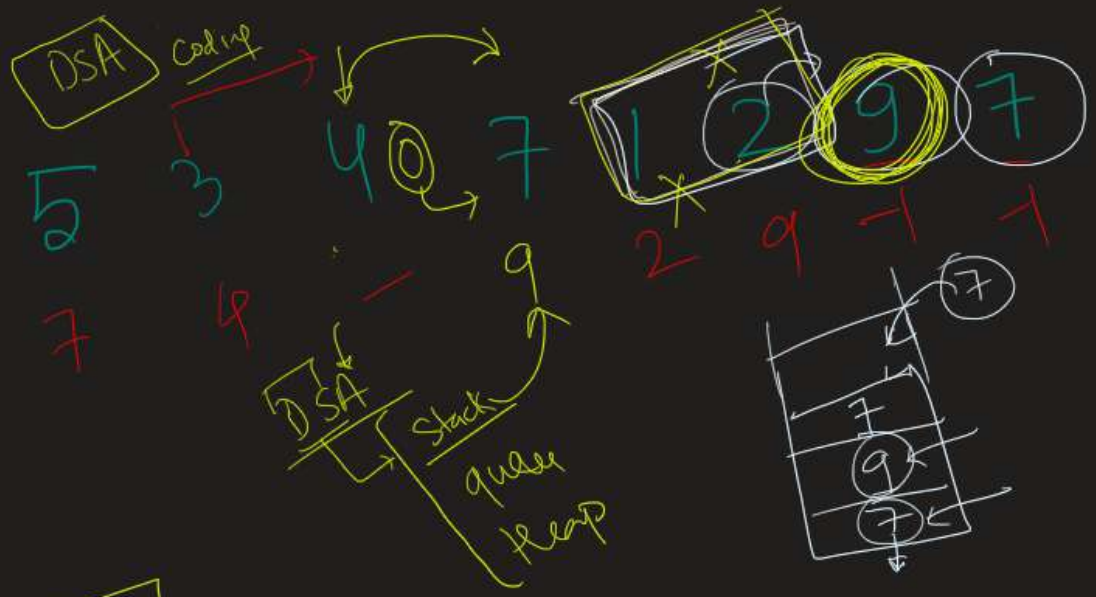


Head
Size = 1

a.next = head
head = a
size ++;

7





word my

CP Word Problem

Next greater element
leetcode

```
#include<bits/stdc++.h>
using namespace std;
```

1.

```
✓ class Node{
    public:
    int data;
    Node *next;
    ✓ Node(int data, Node *next=NULL){
        this->data=data;
        this->next=next;
    }
};
```

```
class Stack{
```

```
    public:
```

```
    int size=0;
```

```
    Node* head;
```

```
    Stack(){
```

```
        size=0;
```

```
        head=NULL;
```

```
    }
```

```
    void push(int num){
```

```
        Node *a = new Node(num);
```

```
        if(size==0){
```

```
            head=a;
```

```
            this->size++;
```

```
        }else{
```

```
            size++;
```

```
            a->next=head;
```

```
            this->head=a;
```

```
        }
```

```
    }
```

2.

```
int top(){  
    return head->data;  
}
```

3.

```
bool isempty(){  
    if(size==0) return true;  
    else return false;  
}
```

```
int pop(){  
    if(size==0) return -1;  
    else{  
        int value=head->data;  
        head = head->next;  
        size--;  
        return value;  
    }  
}
```

```
};
```

4.

```
int main(){  
  
    Stack s;  
    cout<<s.isEmpty()<<endl;  
    s.push(3);  
    s.push(4);  
    s.push(5);  
    s.push(7);  
    s.pop();  
    cout<<s.top()<<endl;  
    cout<<s.isEmpty();  
}
```