

# STACKS

---

Problem Solving with Computers-II

C++

```
#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook\n";
    return 0;
}
```

# Announcements

- Midterm next week (Thursday)!
  - Closed book, closed notes
  - Practice problems available in Canvas
  - Includes all materials covered so far including this week's lectures
  - Data structures covered: Linked lists, BST, stacks and queues
  - Labs 1 - 4 and pa01
  - Leetcode problem sets 1- 3

## Results for **Santa Barbara, CA**

11 PM

Sun



59° 55°

2 AM

Mon



59° 51°

5 AM

Tue



58° 45°

8 AM

Wed



59° 45°

11 AM

Thu



62° 44°

2 PM

Fri



61° 42°

5 PM

Sat



63° 42°

8 PM

Sun



65° 43°

<https://leetcode.com/problems/daily-temperatures/>

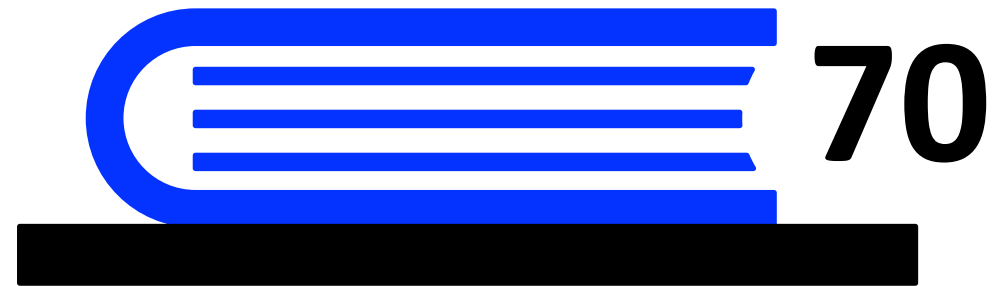
```
stack<int> s
```

**Empty stack**



Operations: **push()**    **pop()**    **top()**

```
stack<int> s  
s.push(70)
```



Operations: **push()**    pop()    top()

```
stack<int> s
```

```
s.push(70)
```

```
s.push(50)
```



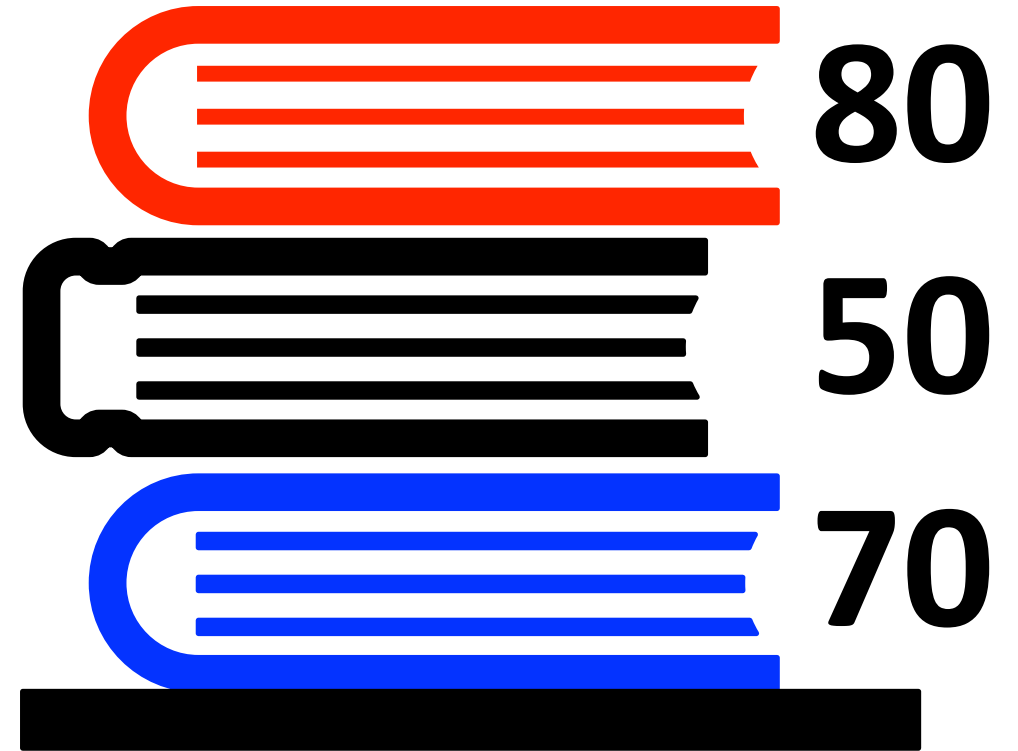
Operations: **push()**    **pop()**    **top()**

```
stack<int> s
```

```
s.push(70)
```

```
s.push(50)
```

```
s.push(80)
```



Operations: **push()**    **pop()**    **top()**

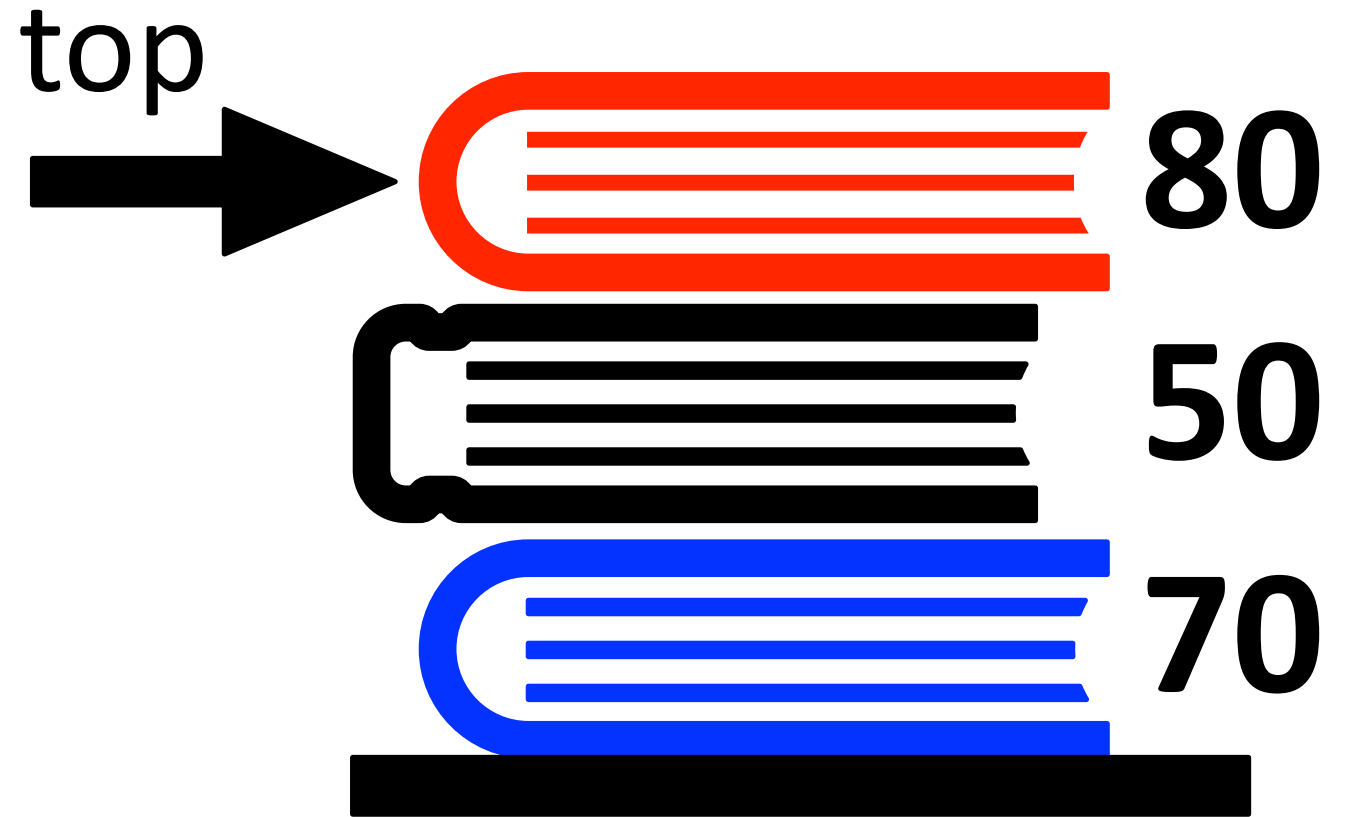
```
stack<int> s
```

```
s.push(70)
```

```
s.push(50)
```

```
s.push(80)
```

**s.top() returns 80**



Operations: `push()`    `pop()`    `top()`



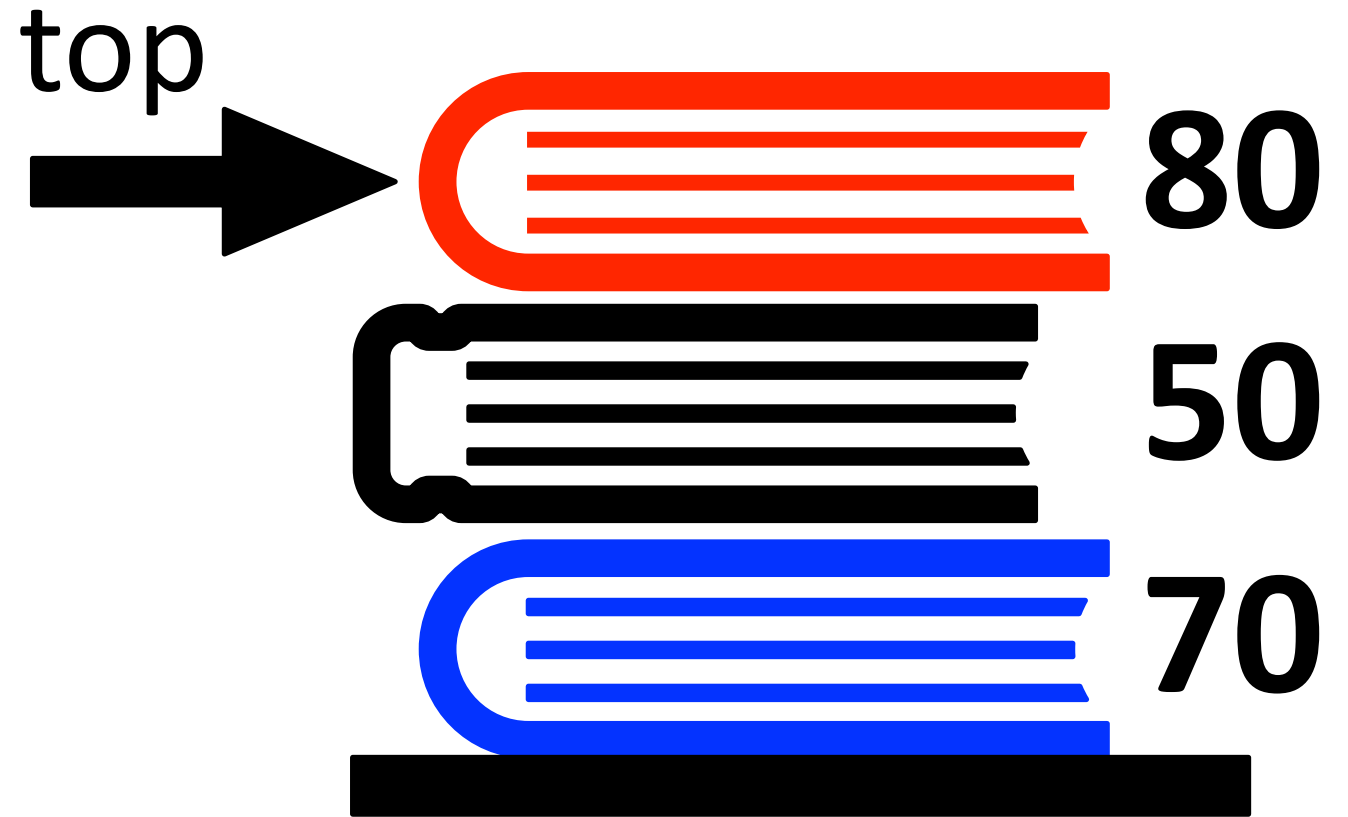
```
stack<int> s
```

```
s.push(70)
```

```
s.push(50)
```

```
s.push(80)
```

```
s.top()
```



**s.pop()** removes value that was pushed in *last*

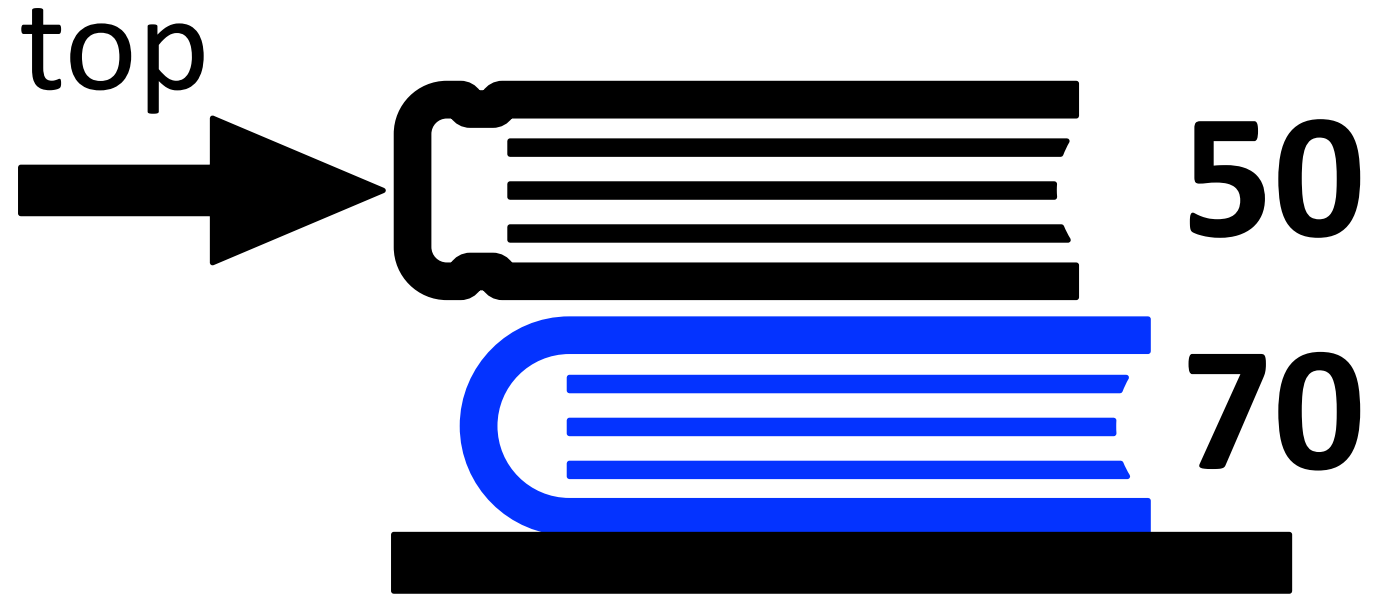
```
stack<int> s
```

```
s.push(70)
```

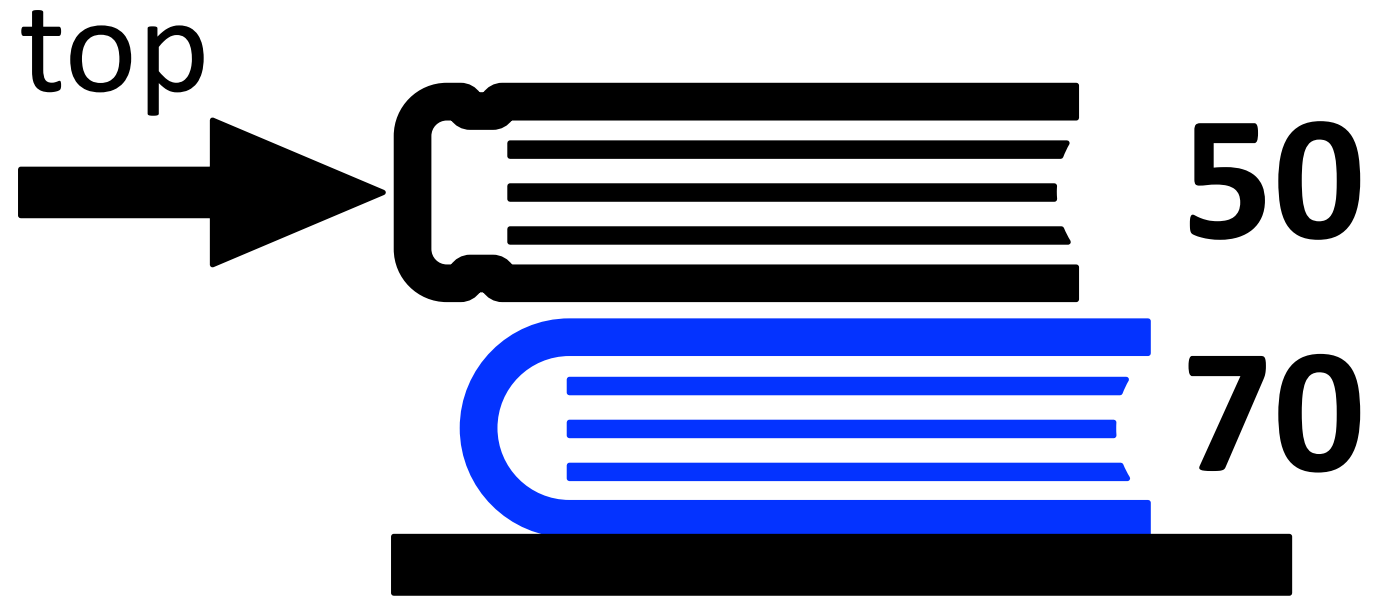
```
s.push(50)
```

```
s.push(80)
```

```
s.top()
```



**s.pop()** removes value that was pushed in *last*



**The Last value In is the First value Out (LIFO)**

## The call stack:

```
1  #include <iostream>
2  using namespace std;
3
4  int fact(int n){
5      if(n <= 1) return 1;
6      return n * fact(n - 1);
7  }
8
9  int main() {
10     cout<< fact(4) << endl;
11     return 0;
12 }
```

main

fact(int)

n | int  
4

fact(int)

n | int  
3

fact(int)

n | int  
2

fact(int)

n | int  
1

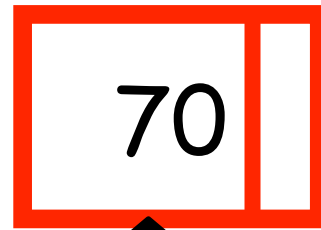
**The Last value In is the First value Out (LIFO)**

**Implement using vector or linked list**

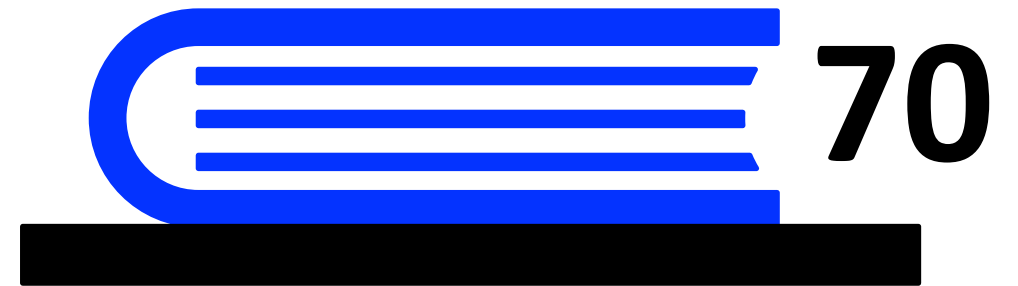


Empty stack

**Stack Abstract Data Type**

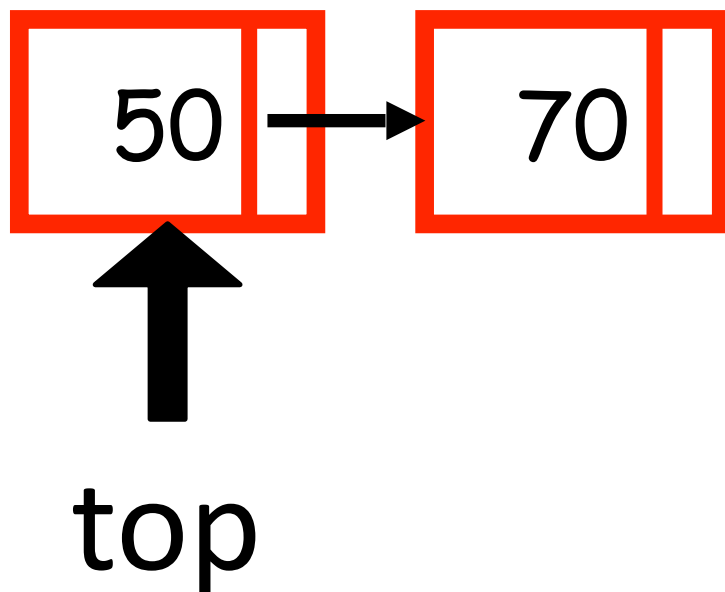


↑  
top



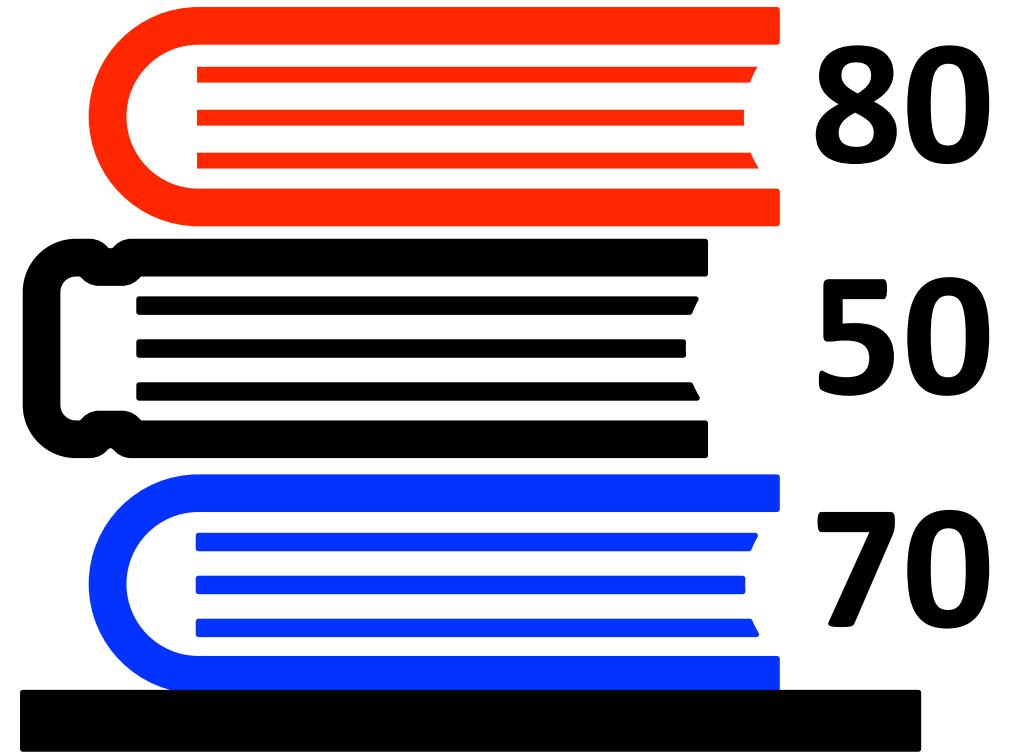
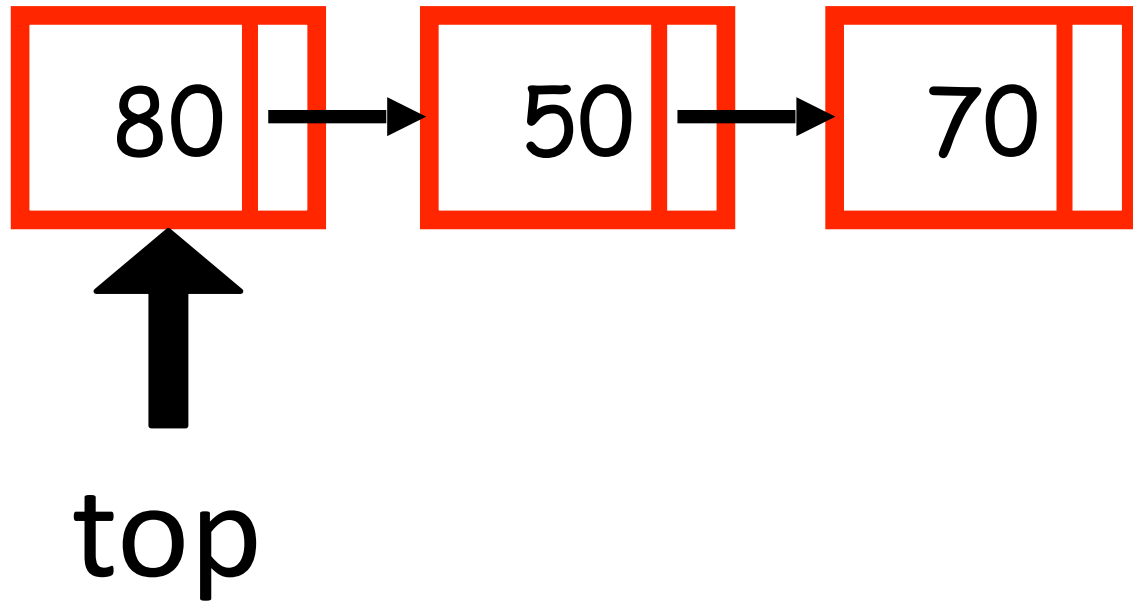
**s.push(70)**

**Stack Abstract Data Type**



**s.push(50)**

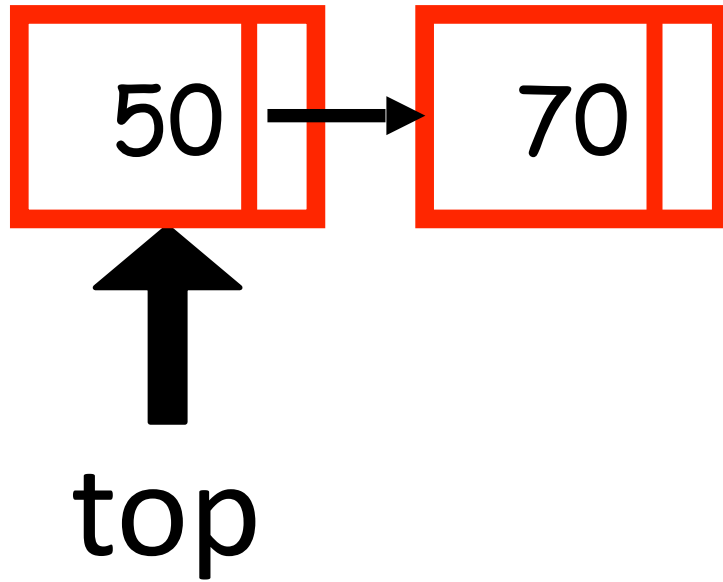
**Stack Abstract Data Type**



**s.push(80)**

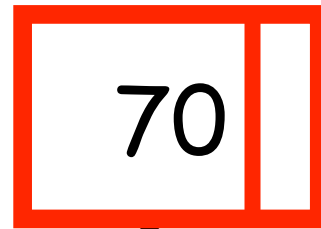
**Stack Abstract Data Type**



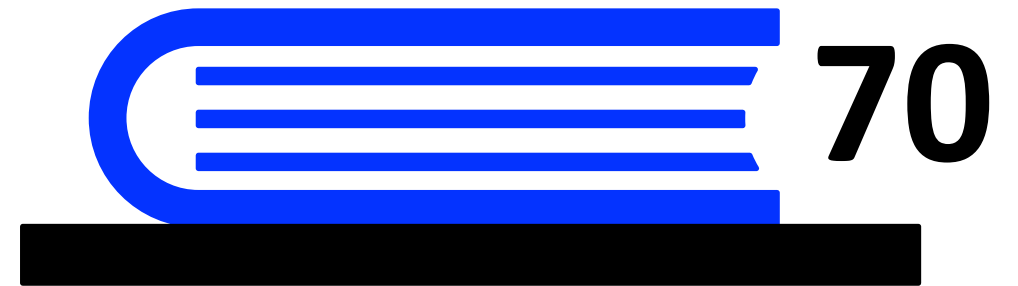


**s.pop()**

**Stack Abstract Data Type**



↑  
top



s.pop()

**Stack Abstract Data Type**

Why implement a stack at all?

After all a stack is a vector or linked list with a  
**reduced set of operations**

Stack has only three operations: **push()**    **pop()**    **top()**

Why implement a stack at all?

After all a stack is a vector or linked list with a  
**reduced set of operations**

A stack is useful for keeping track of history information where computation only depends on the most recent information !!

Stack has only three operations: **push()**    **pop()**    **top()**

Sun



Mon



Tue



Wed



Thu



Fri



Sat



Sun



59

59

58

59

62

61

63

65

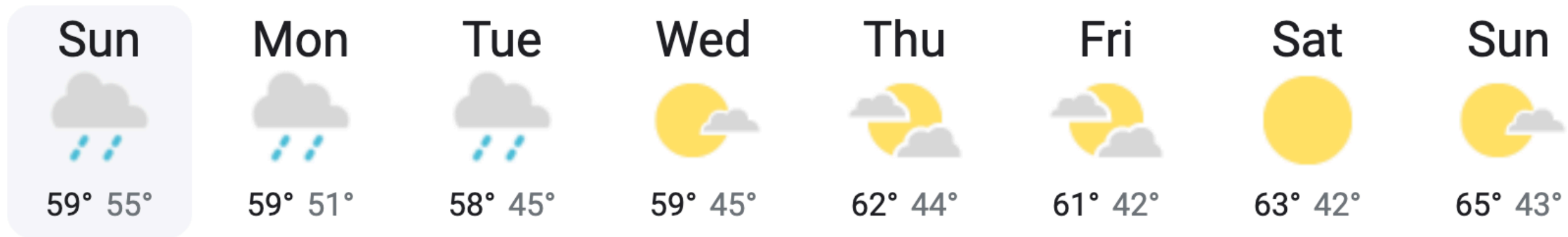
Given an array of integers temperatures represents the daily temperatures, return an array answer such that answer[i] is the number of days you have to wait after the i<sup>th</sup> day to get a warmer temperature. If there is no future day for which this is possible, keep answer[i] == 0 instead.

Input: temperatures = [59,59,58,59,62,61,63,65]

Output: answer = [4, 3, 1, 2, 2, 1, 1, 0]

<https://leetcode.com/problems/daily-temperatures/>

- **Attempt a different solution to this problem on leetcode**
- **Discuss your solutions with the course staff in office hours**



A stack is useful for keeping track of history information where computation only depends on the most recent information !!

<https://leetcode.com/problems/daily-temperatures/>