Data Structures

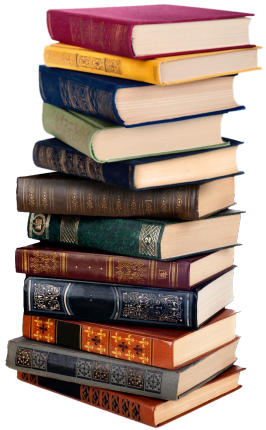
Stack Data Structure

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

What is a stack?

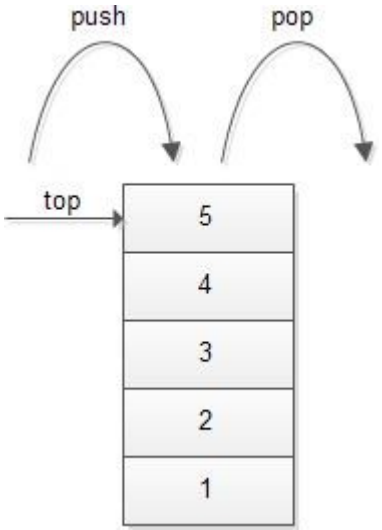
- In English, stack is a pile of objects
  - Stack of boxes
  - Stack of plates
  - We **can stack** books such as in the image
- Let's say we stacked 20 books
  - I asked you to get a book, which one you can easily unstack (remove)?
  - The last (**top**) one?
  - When can we remove the very first one? After all above is removed
  - We call this: **FILO = First in, Last out**



Img src

Stack ADT

- The stack FILO is very common in practice, so we better have a DS for it
- Possible functionalities? We want to add & remove elements
- **push**(element): Add to the **top** of stack
- **pop**(): Remove the **top** of the stack
- **peek**(): What is the top, without removing
- Useful functionalities:
  - Is Empty? Is Full? Clear content?
- Any implementation that satisfies this (FILO) = Stack



Img src

Stack ADT

- Let’s trace the following operations
- Initially empty stack!

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8
- Push 4

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8
- Push 4
- Peek? 4

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8
- Push 4
- Peek? 4
- Pop

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8
- Push 4
- Peek? 4
- Pop
- Push 7

Stack ADT

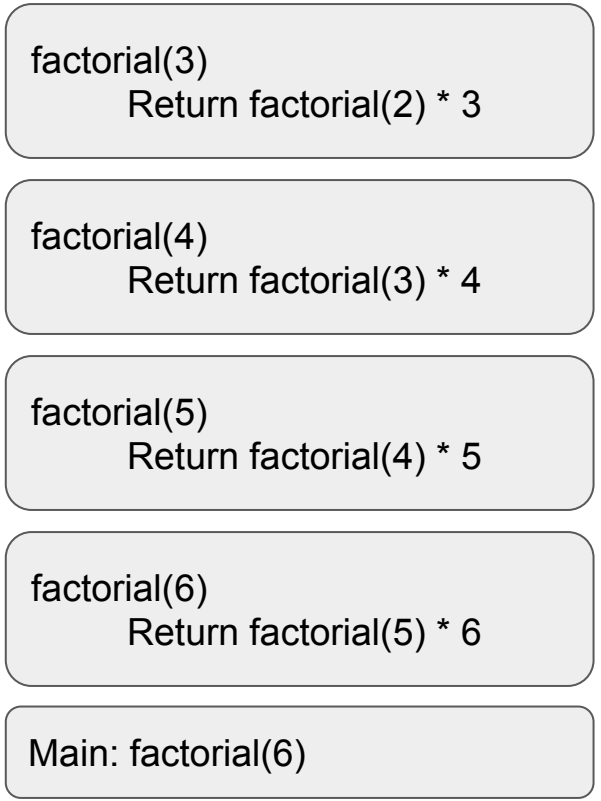
- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8
- Push 4
- Peek? 4
- Pop
- Push 7
- Push 15

Stack ADT

- Let’s trace the following operations
- Initially empty stack!
- Push 5
- Push 8
- Push 4
- Peek? 4
- Pop
- Push 7
- Push 15
- Is Empty? False

Stack: Function Calls

- When function A calls B, which calls C, internally this is a stack of function calls (A, B, C).
  - Function C, the top, must be done first
  - Then goes back to B
  - Finally, A, which is the first, must finish last
- Similarly in recursion.
- For a factorial (6), we expect several calls



Stack: Recursion Trace

- Call **Factorial**(6)
  - If 6 == 1? False
  - Call **Factorial** (5) and multiply results with 6
    - If 5 == 1? False
    - Call **Factorial** (4) and multiply results with 5
      - If 4 == 1? False
      - Call **Factorial** (3) and multiply results with 4
        - If 3 == 1? False
        - Call **Factorial** (2) and multiply results with 3
          - If 2 == 1? False
          - Call **Factorial** (1) and multiply results with 2
            - If 1 == 1? True
              - Return 1

```
int factorial(int n) {
    if (n == 1)
        return 1;
    return factorial(n-1) * n;
}
```

Stack Implementation

- We know now what is Stack & its ADT
- We know we have 2 memory models: Array and Linked List
- We can implement a stack using either way
- Both implementations are simple and intuitive!
    - Give a trial by yourself!

Stack Usage: Array based

- Here, we create a stack of 3 elements
- Push back 3 numbers: Stack is: 10 20 30 (top)
- It is now full, with peek = 30

```
Stack stk(3);
stk.push(10);
stk.push(20);
stk.push(30);
cout<<stk.isFull()<<"\n";    // 1
cout<<stk.peek()<<"\n";      // 30
stk.pop();
cout<<stk.isFull()<<"\n";    // 0
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."