STACK DATA STRUCTURE

Stack and its characteristics

A linear data structure that follows a specific order for operations. Uses LIFO (Last In, First Out) principle. Example: Stack of books → the book placed last is removed first.

Characteristics:

- Access only to the top element.
- Simple and efficient structure.
- Operations take O(1) time.
- Can be implemented using arrays or linked lists.

Stack Operations Overview

push(x):

Insert element x on the top

pop():

Remove element from the top

peek() / top():

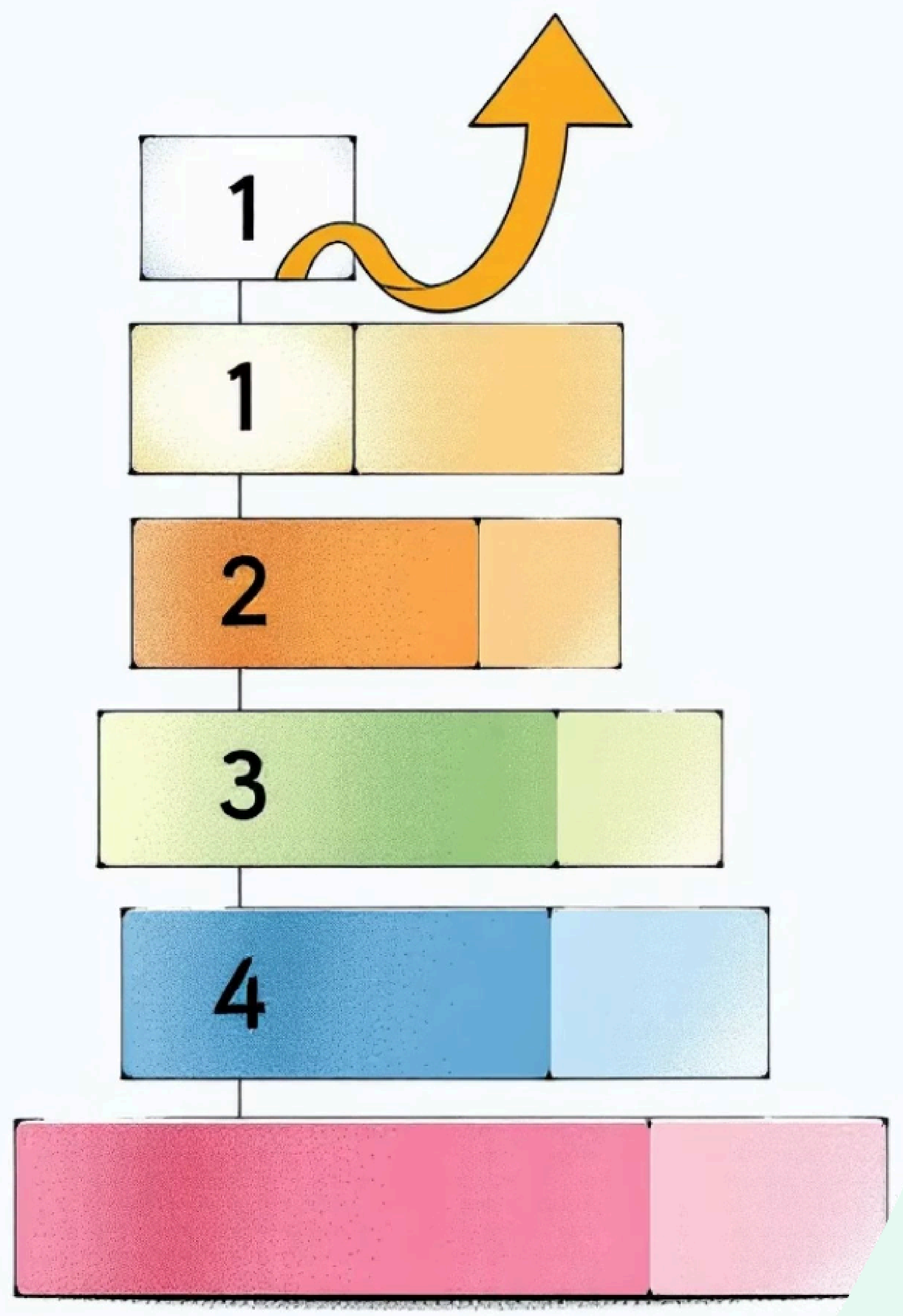
View top element without removing

isEmpty():

Check if stack has no elements

isFull():

(For array implementation) Check if stack is full



Push Operation:

Definition: Adds an element to the top of the stack.

Steps:

- Check if stack is fullIncrement top
- Insert the new element

Time Complexity: O(1)

Push Operation - Step by Step

1

2

3

1. Initial Stack

The stack begins with existing elements. The top element is clearly visible and accessible.

2. Adding New Element

A new element is introduced, positioned directly above the current top of the stack.

3. Updated Stack

The new element is now successfully added and becomes the new top of the stack, ready for the next operation.

Pop Operation

Definition: Removes the top element.

Steps:

- Check if stack is empty
- Return the element at top
- Decrement top

Time Complexity: O(1)

Pop Operation - Step by Step

1

2

3

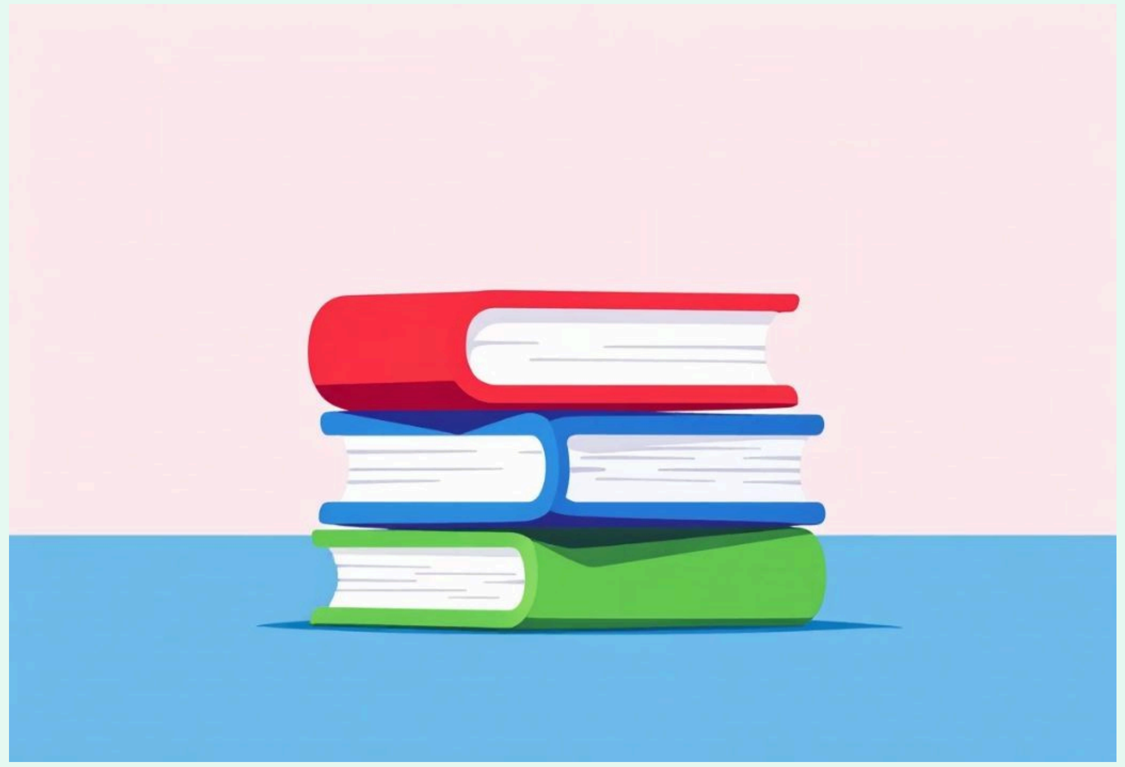
1. Initial Stack

The stack begins with existing elements. The top element is clearly visible and accessible.

2. Removing Element

The top element is carefully lifted from the stack, adhering to the LIFO principle.

3. Updated Stack



The stack is now updated with one less element, and the new top element is ready for subsequent operations.

Peek Operation

Returns the top element without removing it. Useful for inspecting the current top value.

Time Complexity: O(1)

Applications of Stack

Expression evaluation:

Postfix, prefix, infix

Function call stack:

Maintains function calls in runtime

Undo/Redo operations

(editors)

Backtracking algorithms

Parenthesis matching

DFS (Depth First Search)

Advantages & Disadvantages

Advantages

- Easy to implement
- Fast operations
- Useful in many system-level tasks

Disadvantages

- Limited size (if array-based)
- Access allowed only at one end

Conclusion:

Stack uses LIFO Core operations:

push, pop, peek Implemented using arrays/linked lists Widely used in algorithms, compilers, and system architecture

THANK YOU!

Team Members

AP24110011915 - R. Vidya

AP24110011926 - K. Jeevana Sarayu

AP24110011931 - V. Jeeshitha Sai

AP24110011934 - R. Alpana

AP24110011935 - G. Chaitanya Siri

AP24110012176 – Rolli Gupta