Stack – Stack Operations

1. Introduction

Stack operations are the basic actions performed on a stack data structure.

All operations follow the **LIFO (Last In, First Out)** principle and are performed **only at one end**, called the **Top** of the stack.

Understanding stack operations is essential before implementing stacks using arrays or linked lists.

---

2. Main Stack Operations

A stack supports the following core operations:

1. **Push**
2. **Pop**
3. **Peek (Top)**
4. **isEmpty**
5. **isFull** (for array-based stack)

---

3. Push Operation

What is Push?

Push is the operation of **adding an element to the top of the stack**.

Logic (Plain English)

1. Check if the stack is full
2. If full, report stack overflow
3. Otherwise, increase the top pointer
4. Insert the new element at the top position

Example

```
Push 10
Push 20
Push 30
```

Stack:

```
Top → 30
      20
      10
```

---

4. Pop Operation

What is Pop?

Pop is the operation of **removing the top element from the stack**.

Logic (Plain English)

1. Check if the stack is empty
2. If empty, report stack underflow
3. Otherwise, remove the top element
4. Decrease the top pointer

Example

```
Pop → removes 30
```

Stack becomes:

```
Top → 20
      10
```

---

5. Peek (Top) Operation

What is Peek?

Peek returns the **top element of the stack without removing it**.

Logic (Plain English)

1. Check if the stack is empty
2. If empty, no element to show
3. Otherwise, return the top element

Example

```
Peek → 20
```

Stack remains unchanged.

---

6. isEmpty Operation

What is isEmpty?

isEmpty checks whether the stack has **no elements**.

Logic

- If top == −1 → Stack is empty
- Otherwise → Stack is not empty

This operation is commonly used before **pop** or **peek**.

---

7. isFull Operation

What is isFull?

isFull checks whether the stack is **completely filled** (array implementation).

Logic

- If top == size − 1 → Stack is full
- Otherwise → Stack has space

This operation is used before **push**.

---

8. Stack Overflow

What is Stack Overflow?

Stack overflow occurs when:

- A push operation is performed on a **full stack**

Example

```
Push when top == size − 1
```

This causes an error condition.

---

9. Stack Underflow

What is Stack Underflow?

Stack underflow occurs when:

- A pop operation is performed on an **empty stack**

Example

```
Pop when top == −1
```

This is also an error condition.

---

10. Time Complexity of Stack Operations

Operation	Time Complexity
Push	O(1)

Operation	Time Complexity
Pop	O(1)
Peek	O(1)
isEmpty	O(1)
isFull	O(1)

All stack operations are **constant time**.

---

11. Advantages of Stack Operations

- Very fast execution
- Simple logic
- Efficient memory usage
- Ideal for recursive and backtracking problems

---

12. Limitations of Stack Operations

- No random access
- Fixed size in array implementation
- Overflow and underflow must be handled carefully

---

13. Real-World Applications of Stack Operations

- Function call management
- Undo and redo features
- Expression evaluation
- Syntax checking
- Browser history navigation

---

14. Summary

- Stack operations work only at the top
- Push adds, Pop removes
- Peek views without removing
- isEmpty and isFull handle safety
- All operations run in O(1) time
