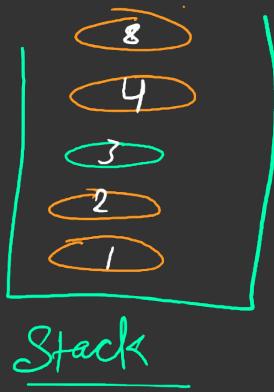
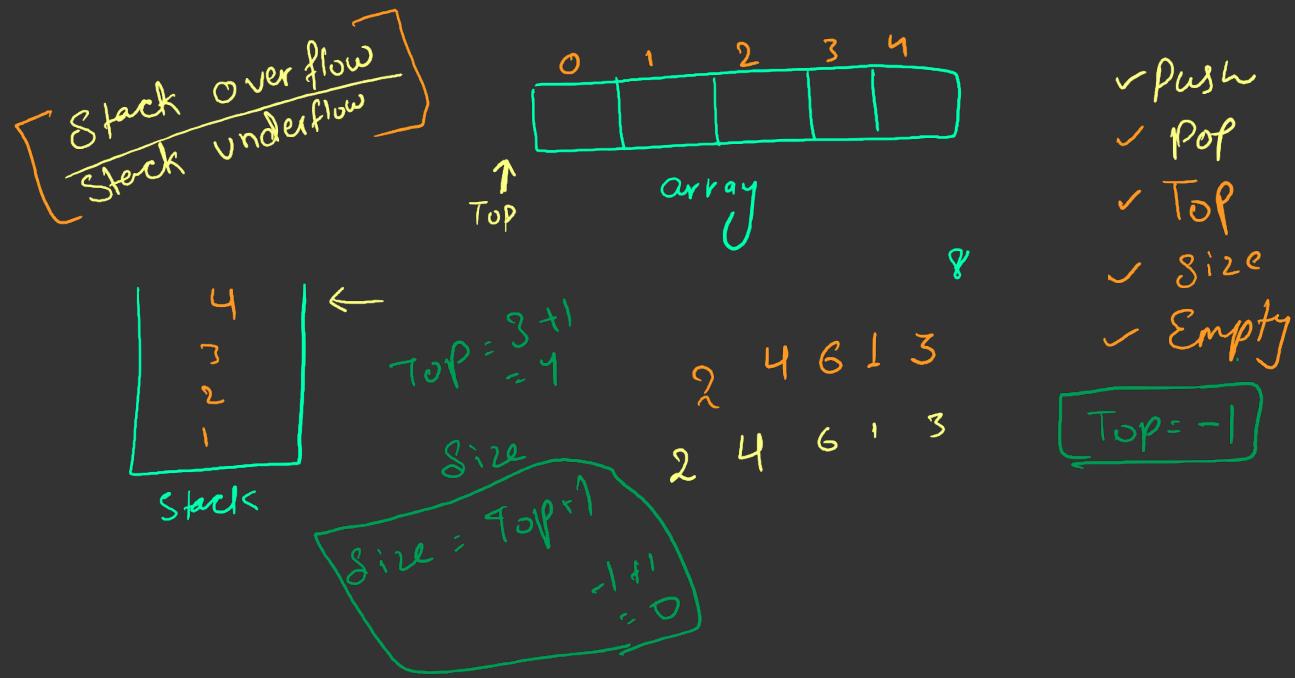


## Introduction to Stack

- ✓ Push operation
- ✓ Pop operation - dd
- ✓ Top operation
- ✓ Size operation
- ✓ Empty operation

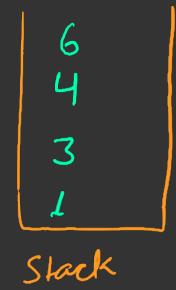


- ✓ LIFO
- ✓ FIFO

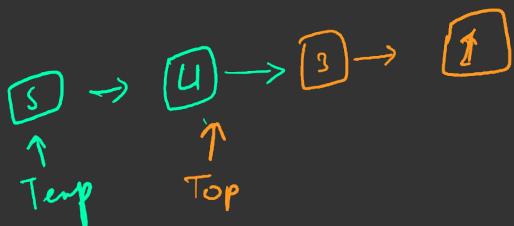


```
class Stack {
    int *arr;
    int size;
    int top;
public:
    Stack(int s) {
        size = s;
        top = -1;
        arr = new int[s];
    }
    void Push(int value) {
        if (Top == size - 1) {
            cout << "Stack overflow";
            return;
        } else {
            Top++;
            arr[Top] = value;
        }
    }
    void Pop() {
        if (top == -1) {
            cout << "Stack underflow";
            return;
        } else {
            Top--;
        }
    }
}
```

$S_{12} \rho$



1 3 4 6 8



size: 84

pop  
size: 74

```

class Node {
public:
    int data;
    Node *next;
    Node (int val) {
        data = val;
        next = NULL;
    }
}

```

```

class STACK2 {
    Node *TOP;
    int size;
public:

```

{Stack overflow}

```

    Stack() {
        TOP = NULL;
        size = 0;
    }
    void Push (int val) {
        Node *temp = new Node (val);
        temp->next = TOP;
        TOP = temp;
        size++;
    }
    void Pop() {
        if (TOP == NULL) {
            cout << "underflow";
            return;
        }
        Node *temp = TOP;
        TOP = TOP->next;
        delete Temp;
    }
}

```

## STL

```
↳ Stack <int> S;  
S.push(5);  
S.push(10);  
S.pop();
```

Heap

deque

