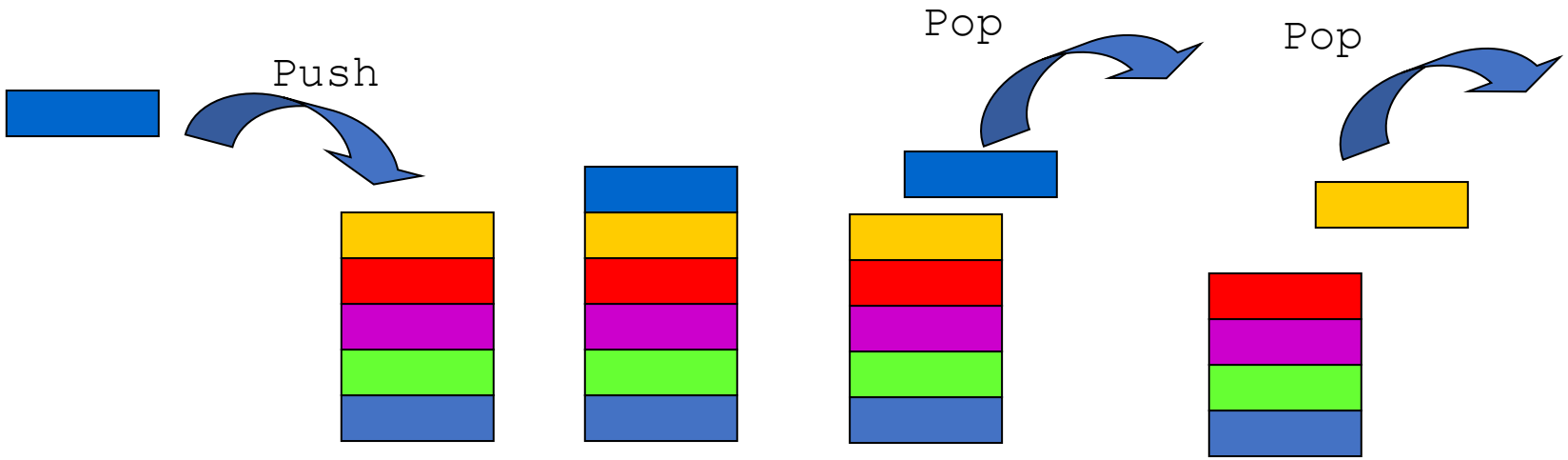
Stack DS

Introduction

- A list for which Insert and Delete are allowed only at one end of the list (the *top*)
  - **LIFO – Last in, First out**



What is Stack Good For?

- Page-visited history in a Web browser
- Undo sequence in a text editor

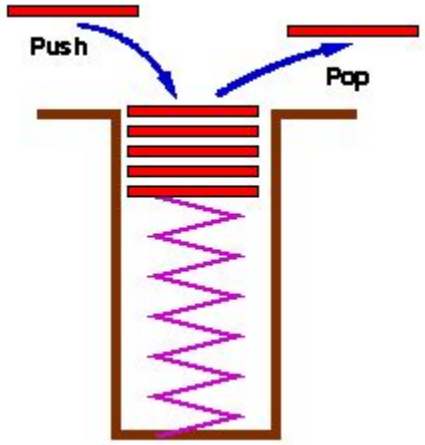
Stack ADT

Objects:

A finite sequence of nodes

Operations:

- Create: Create a list for stack and decide the size
- Push: Insert element at top
- Top / Peek: Return top element
- Pop: Remove and return top element
- isFull: test if stack is full
- isEmpty: test if stack is empty



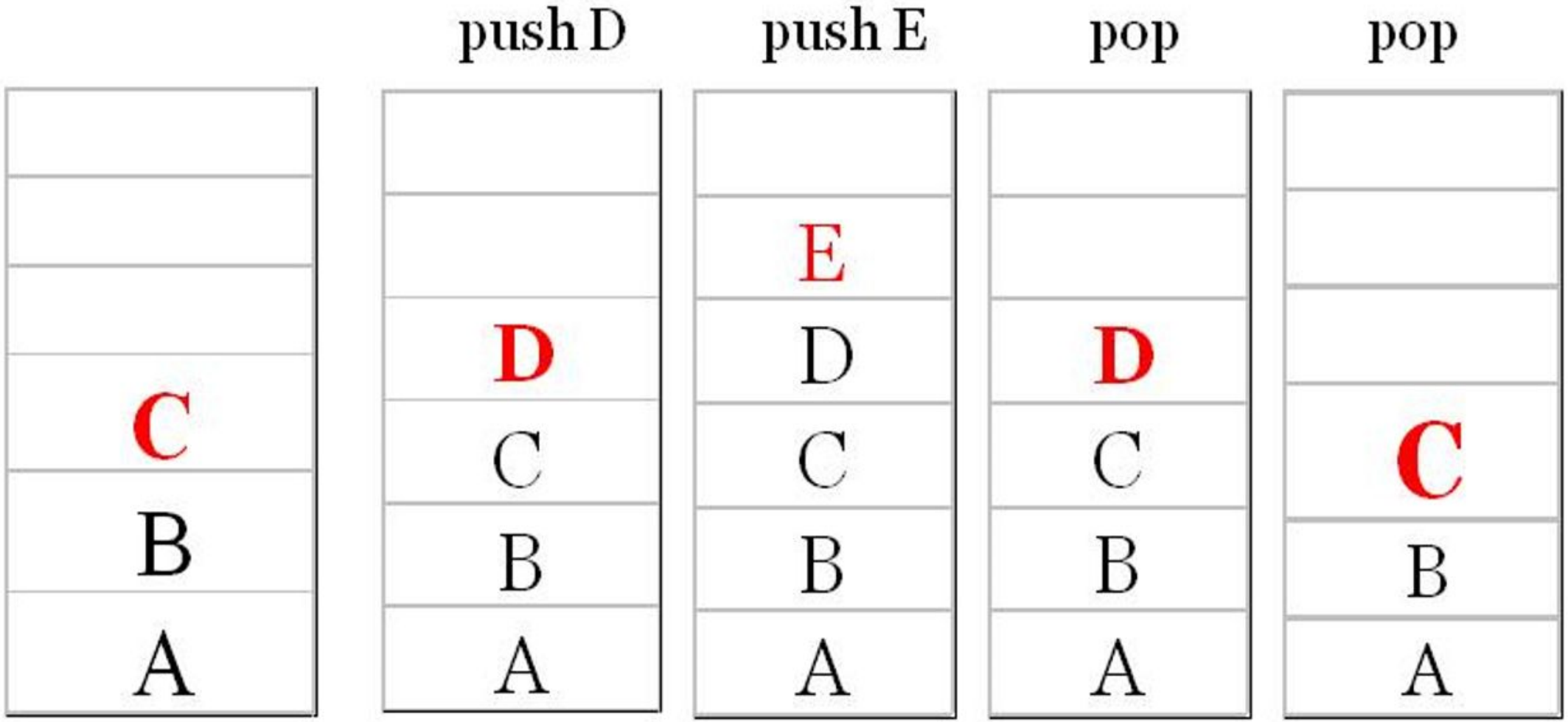
Stack – Implementation

- Initialization:
  - Top = -1 (To have the index of the last element inserted)
  - Max = n (To fix the size of the stack)

```
isFull(stack){
    return (top == MAX-1)
}
```

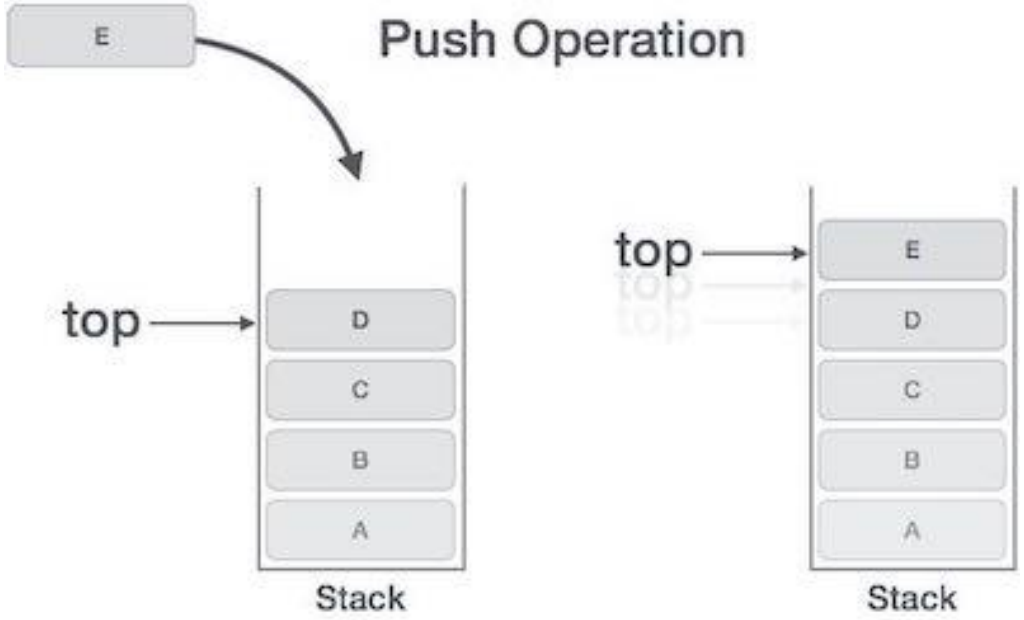
```
isEmpty(stack){
    return (top==-1)
}
```

Push and Pop



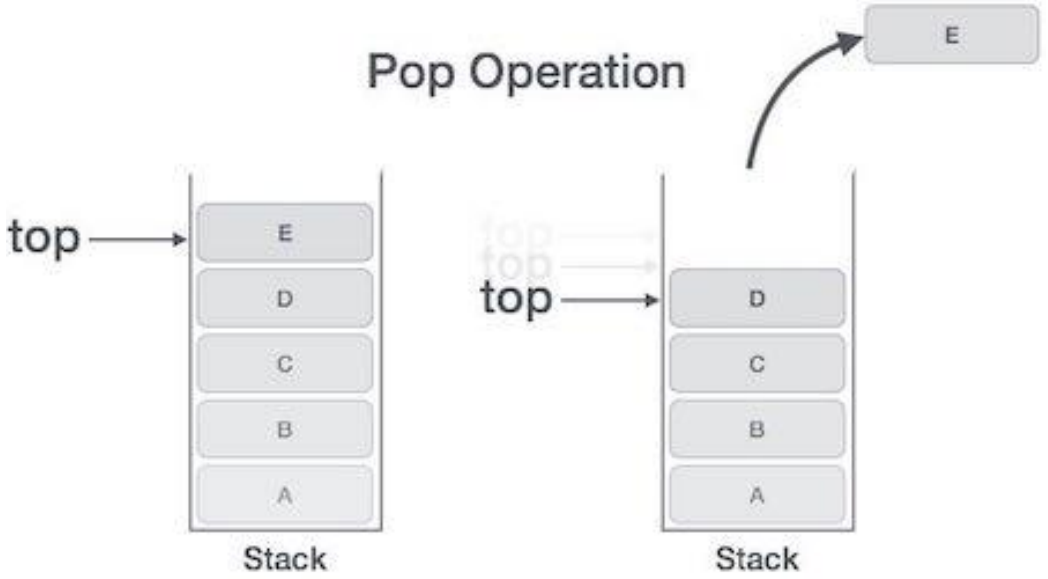
Algorithm for push

```
Push(stack,data){
    if (!isfull(stack))
        top = top + 1
        stack[top] = data
    else
    print("Stack Full")
}
```



Algorithm for Pop

```
Pop(stack){
    if (!isEmpty(stack))
        data ← stack[top]
    top ← top - 1
    return data
     else
    print("Empty Stack")
}
```



Display and Peek Functions

```
Display(stack){
  if(isEmpty())
   cout<<"Stack underflow";
  else
   for(i=0;i<=top;i++)
      cout<<stack[i];
}
```

```
Peek(stack){
  if(isEmpty())
   cout<<"Stack underflow";
  else
   return(stack[top]);
}
```