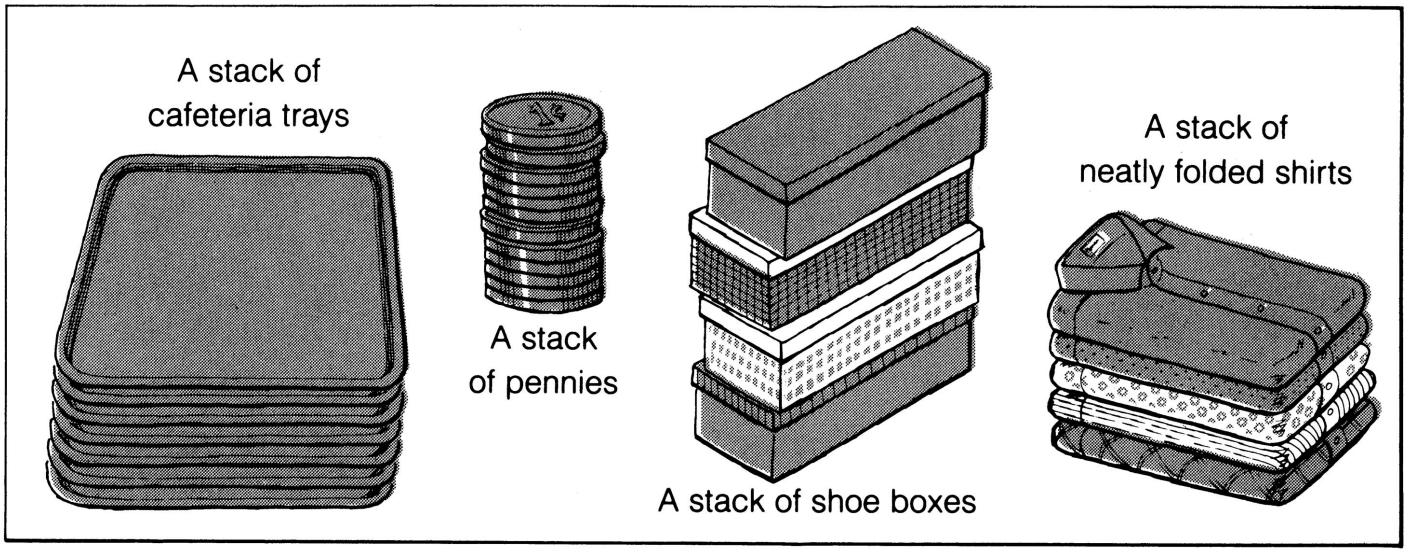
# Stack

# What is a stack?

- A stack is a special type of data structure
- It is an ordered group of homogeneous items of elements.
- Elements are added to and removed from the top of the stack (the most recently added items are at the top of the stack).
- The last element to be added is the first to be removed (**LIFO**: Last In, First Out).

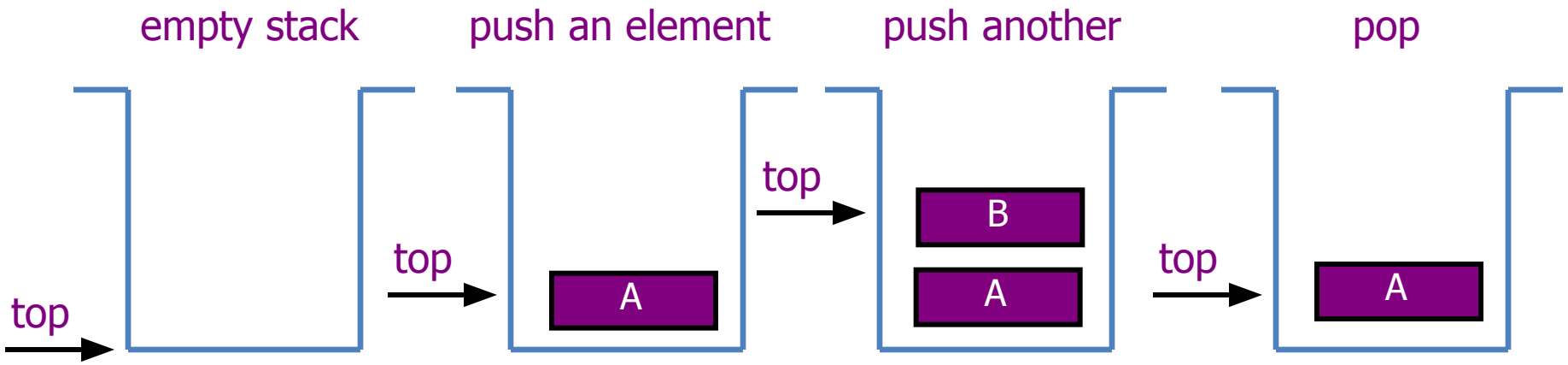


# Stacks

- A stack is a LIFO (Last-In/First-Out) data structure
- A stack is sometimes also called a pushdown store.
- What are some applications of stacks?
  - Conversion of expressions
  - Evaluating of expressions

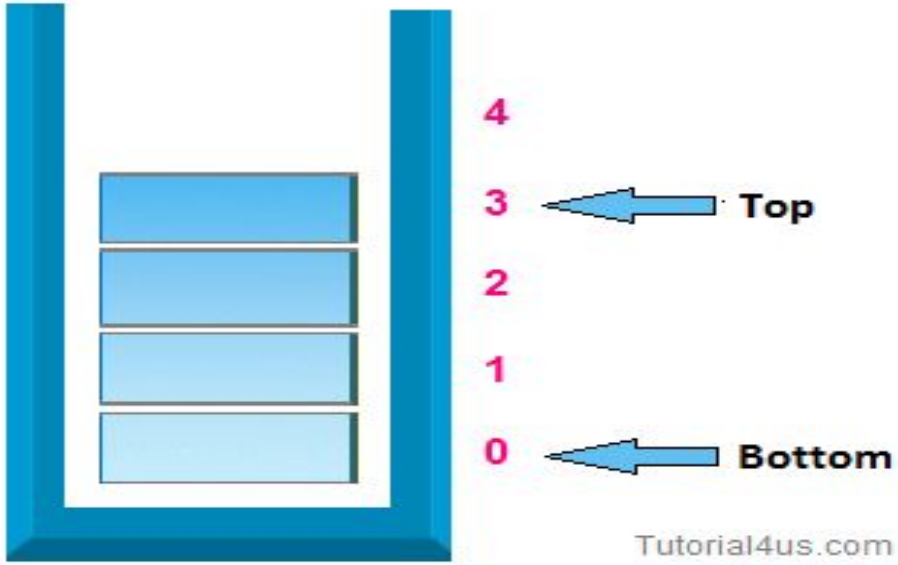
# Operations on stack Push and Pop

- Primary operations: Push and Pop
- Push
  - Add an element to the top of the stack
- Pop
  - Remove the element at the top of the stack



# Push Operation

- Inserting an element into the stack is called push operation.
- Only one item is inserted at a time and item has to be inserted only from top of the stack
- When elements are being inserted there is possibilty of stack being full.
- Trying to insert an element when stack is full results in **overflow of stack**


4
3
Top
2
1
0
Bottom
Tutorial4us.com

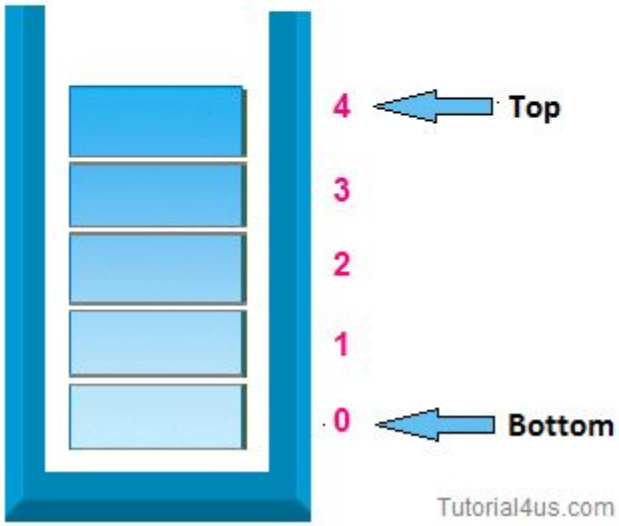
- **Algorithm for push**
- Initialization, set top=-1
- Repeat step 3 to 5 until top<Max size-1
- Read, item
- Set top=top+1
- Set stack[top]=item
- Print "stack overflow"

```
void push()
{
 int item;
 if(top==size-1)
 {
printf("\n stack is full");
}
 else
{
top=top+1;
printf("\n\n Enter element in stack: ");
scanf("%d",&s.item);
s.stack[top]=s.item;
}
}
```

# POP Operation

- In case of stack deletion of any item from stack is called **pop**. In any item is delete from top of the stack, When you delete any item from stack top will be decreased by 1.

- **Algorithm for pop**
- Repeated steps 2 to 4 until top>=0
- Set item=stack[top]
- Set top=top-1
- Print "Item deleted"
- Print "Stack under flow"


4
Top
3
2
1
0
Bottom
Tutorial4us.com

- **Example of Pop Item From Stack**

```
void pop()
 {
int item;
 if(top==-1)
{
printf("\nStack is empty: ");
}
else
{
 item=stack[top];
top=top-1;
printf("deleted data is: %d",item);
}
}
```