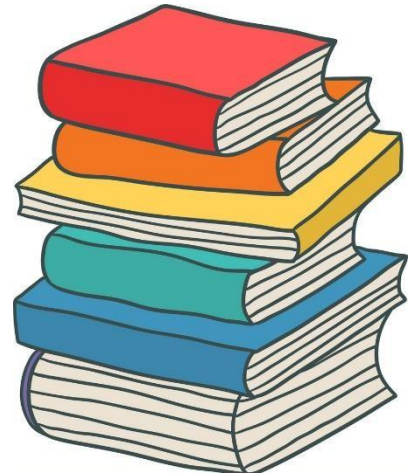
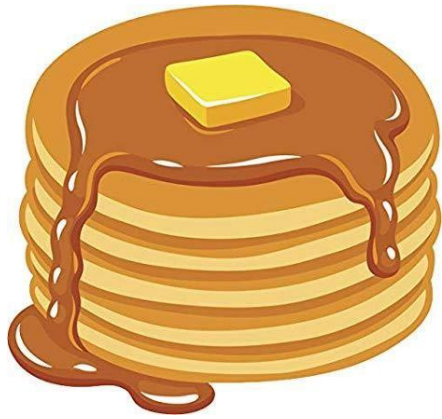


DATA STRUCTURES & ALGORITHMS

Stacks

Instructor: Engr. Laraib Siddiqui

Stack

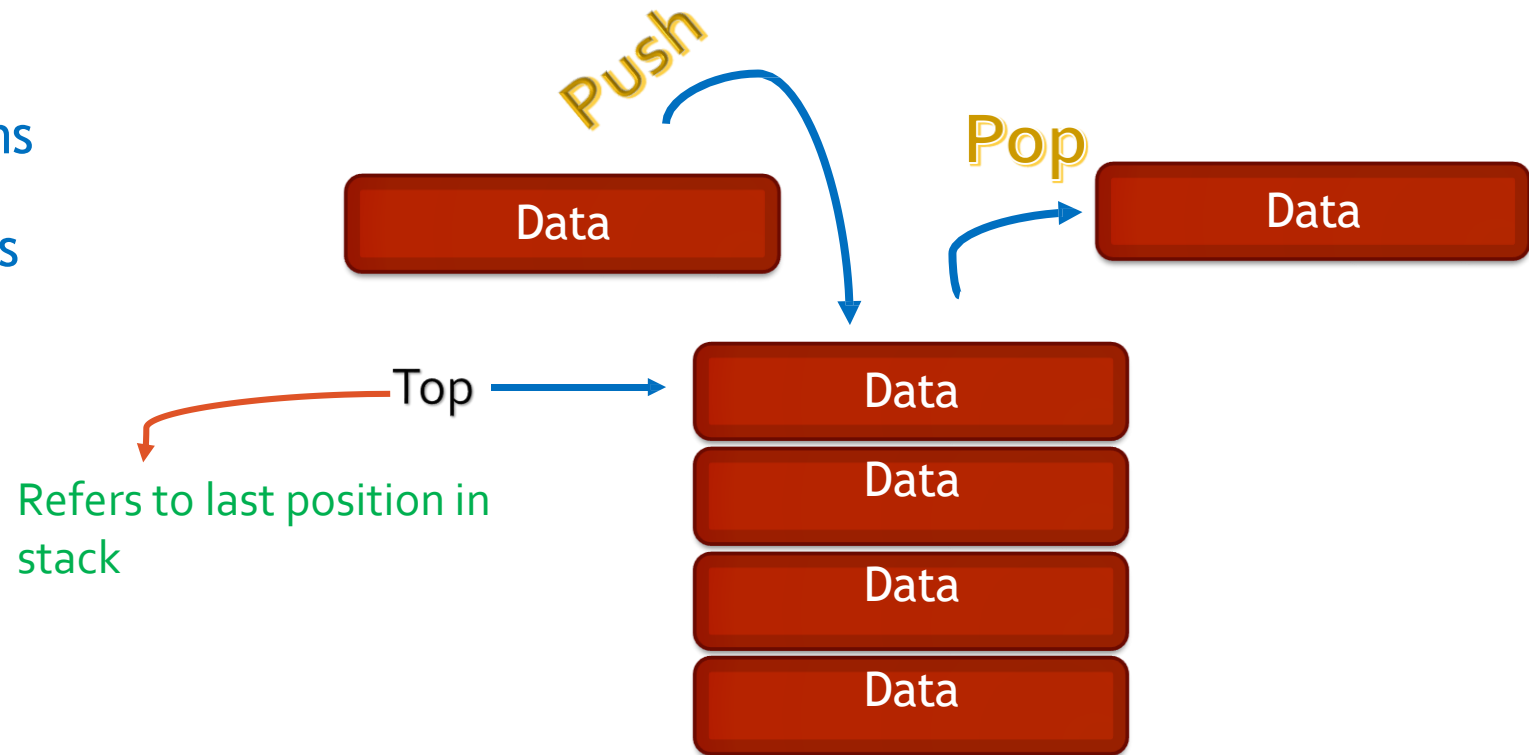


Stack

One ended linear data structure.

Primary operations

- Push -> insert items
- Pop -> delete items



Example

Pop()

Push('Cat')

Push('Dog')

Pop()

Pop()

Pop()

Push('Cat')

Push('Dog')

Pop()

Pop()

Dog

Cat

Lion

Tiger

Zebra

Cat

Lion

Tiger

Zebra

Dog

Cat

Lion

Tiger

Zebra

Lion

Tiger

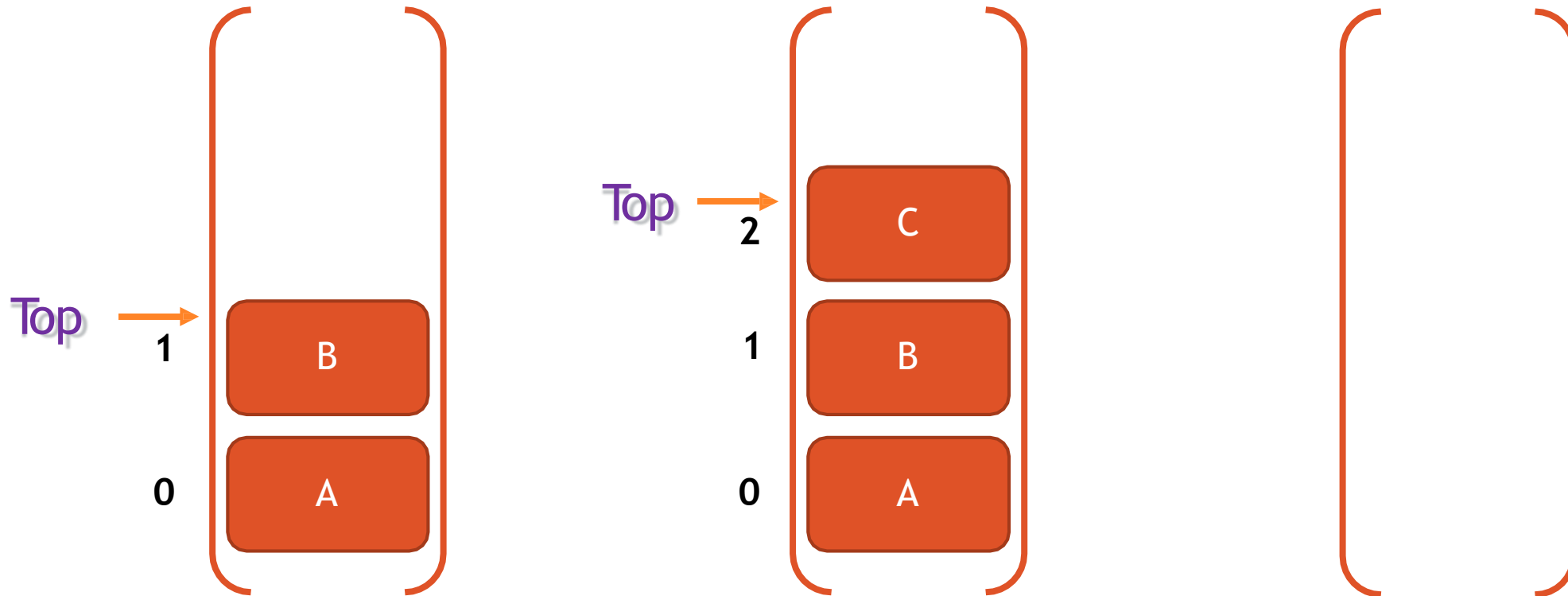
Zebra

Cat

Applications

- Undo mechanisms in text editors.
- Compiler syntax for matching brackets.
- Recursion by keeping track of previous function calls.
- Depth first search on a graph.

Implementation -Arrays



Using Arrays

PUSH (STACK, TOP, MAXSTK, ITEM)

This procedure pushes an ITEM onto a stack.

1. If $TOP = MAXSTK$, then: Print: OVERFLOW and Return. }

Check for availability

2. Set $TOP := TOP + 1$. }

Increasing top

3. Set $STACK[TOP] := ITEM$. }

Inserting new item on top

4. Return.

Using Arrays

POP (STACK, TOP, ITEM)

This procedure deletes the top element of STACK and assigns it to the variable ITEM.

1. If $TOP = 0$, then: Print: UNDERFLOW and Return. }

Check for underflow

2. Set $ITEM := STACK[TOP]$. }

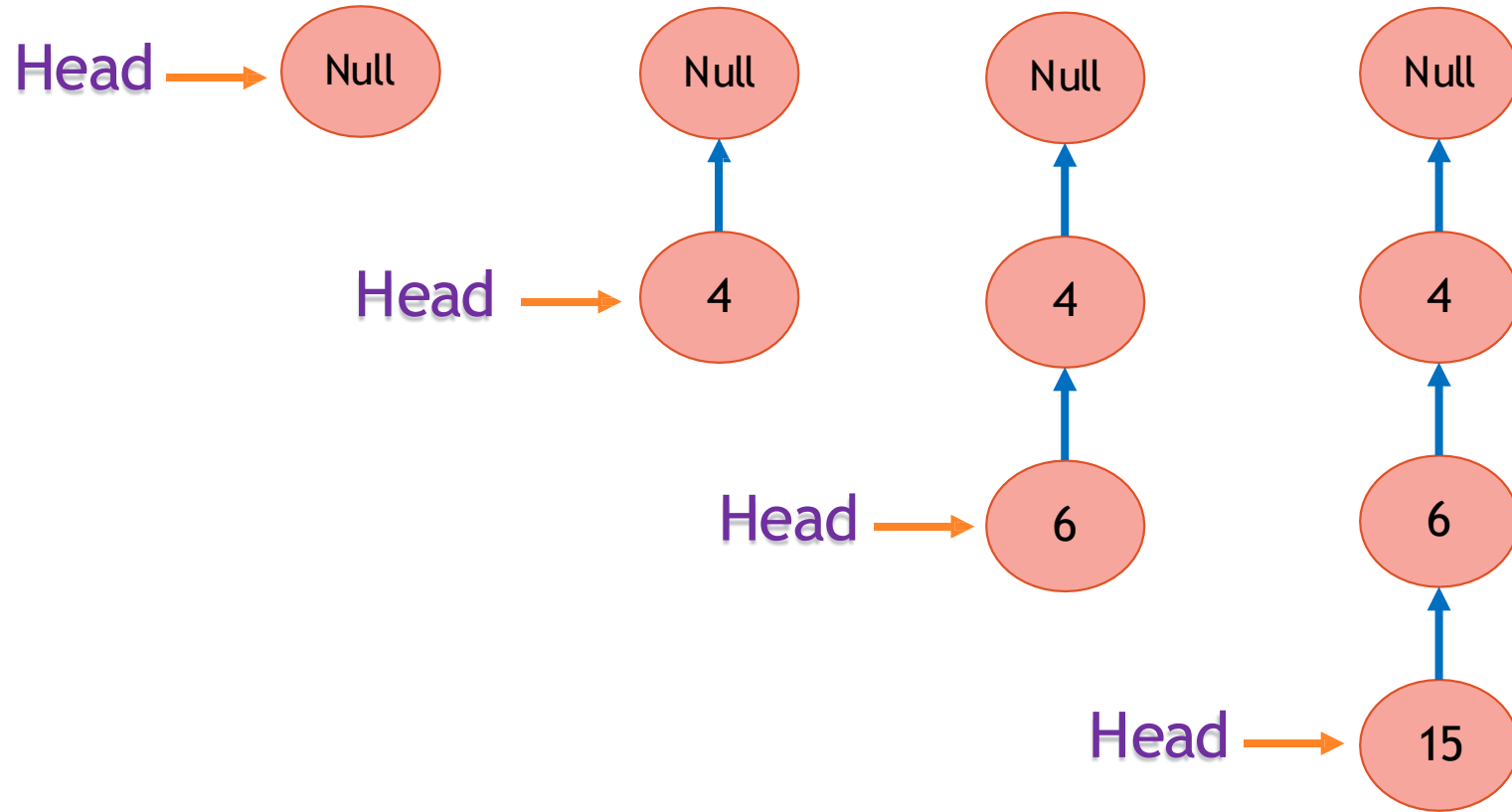
Assigns TOP element to ITEM

3. Set $TOP := TOP - 1$. }

Decreasing top

4. Return.

Implementation– Linked List



Using LinkedList

PUSH_LINKSTACK(INFO, LINK, TOP, AVAIL, ITEM)

This procedure pushes an ITEM into a linked stack

1. If $AVAIL = NULL$, then Write **OVERFLOW** and Exit } Check for availability
2. Set $NEW := AVAIL$ and $AVAIL := LINK[AVAIL]$ } Remove first node from available list
3. Set $INFO[NEW] := ITEM$ } Copies ITEM into new node
4. Set $LINK[NEW] := TOP$ } New node points to original top node
5. Set $TOP = NEW$ } Reset Top to point to new node
6. Exit

Using LinkedList

POP_LINKSTACK(INFO, LINK, TOP, AVAIL, ITEM)

This procedure deletes the top element of a linked stack and assigns it to the variable ITEM

1. If TOP = NULL then Write: UNDERFLOW and Exit. }

Check for underflow

3. Set ITEM := INFO[TOP] }

Copies the top element into ITEM

[Remember the old value of the TOP pointer in TEMP and reset TOP to the point to the next element in the stack]

4. Set LINK[TEMP] = AVAIL and AVAIL = TEMP. }

Returns deleted node to the available list

5. Exit.

Complexity

	Stack
Access	$O(n)$
Search	$O(n)$
Deletion	$O(1)$
Insertion	$O(1)$