

## Algorithms & Data Structures I Week 6 Lecture Note

Notebook: Algorithms & Data Structures I

Created: 2020-10-21 4:14 PM

Updated: 2020-10-29 2:09 PM

Author: SUKHJIT MANN

### Cornell Notes

#### Topic:

Vectors, stacks and queues part 2

Course: BSc Computer Science

Class: CM1035 Algorithms & Data Structures I [Lecture]

Date: October 29, 2020

### Essential Question:

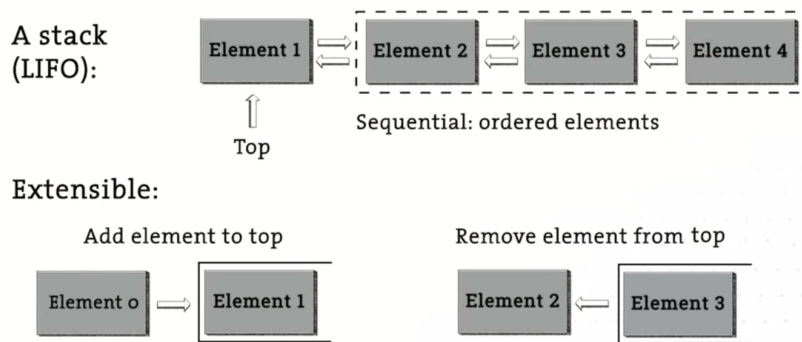
What are Vectors, stacks, and queues?

### Questions/Cues:

- What is a stack?
- What are the operations that we can perform on a stack?

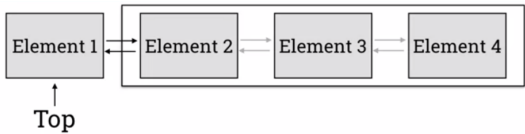
### Notes

- Stack = Similar to a queue, but differs in one important way, in a stack there are no longer two elements that are accessible: the head and the tail in the queue. In a stack, there is only one element that is accessible called the "top" of the stack. Since this is the only element that is accessible, we can only add and remove elements from the top of the stack.
  - In a stack, we have last-in first-out access or LIFO. In other words, a first-in last-out or FILO
  - A stack is for a reliable and ready source of data. We only to access data in one specific location, so we can quickly load and unload data



- Operation(Stack) = The possible operations on a stack are as follows:
  - push![o] operation which adds an element to the top of the stack with o being written onto this element
  - top operation which just returns the top element of the stack, so the value that is stored on the top
  - pop! operation which return the value on the top of the element of the stack and then removes that element from the stack

- empty? operation which just asks if the stack has no elements, returning true if it's indeed empty and false otherwise



Operation	Pseudocode
push! $[o]$	PUSH $[o, s]$
top	TOP $[s]$
pop!	POP $[s]$
empty?	EMPTY $[s]$
Construct new (empty) stack	new Stack $s$

Summary

In this week, we learned about what a stack is and the operations we can perform on a stack.