Lecture 8

Christopher Godley

CSCI 2270 Data Structures

June 12th, 2018

Linked Lists: Complexity

- Insert
  - Head/Tail: O(1)
  - End of singly linked: O(n)
  - Middle/Overall: O(n)
- Search
  - O(n)
- Delete
  - Head/Tail: O(1)
  - Middle/Overall: O(n)

Stacks

- Stores a collection of elements and restricts which element may be accessed at any time.
- Operate on a last in, first out principle (LIFO)
- Think about a *stack* of plates
  - You have 3 plates stacked on top of each other
  - To place a new plate onto the stack, it must go on top
  - To remove a plate from the stack, it must come from the top

Stacks

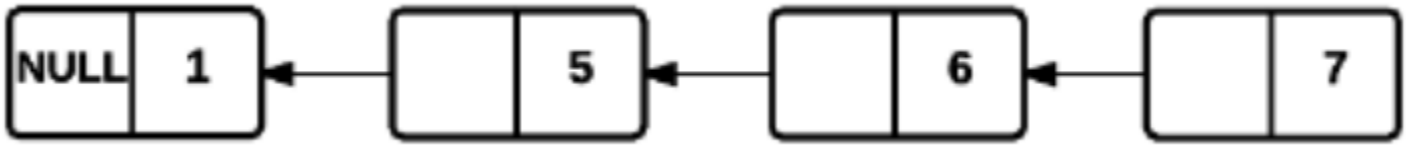
- Placing a new element onto the stack is called a **push**
- Removing an element from the stack is called a **pop**
- Think about reading a sentence
  - Push each word you read onto the stack
  - After reading the whole sentence, pop each word from the stack
  - Compare this to the original sentence

Stacks: Implementation

- Array
  - An array can be turned into a stack by implementing restrictions on where you can add/remove elements
  - The “top” of a stack implemented from an array is set to be the *numElements-1* index
    - Why?

Stacks: Implementation

- Linked Lists
  - Linked lists may also be used to implement stacks
  - Each node represents a data element of the stack
  - Each node stores a pointer to the *prev* node in the list
    - The bottom of the list has a *prev* pointer to NULL
  - The *top* of the stack is a pointer to a node



Stack: ADT

Stack:

- private:
  - top
  - data
  - maxSize
- public:
  - Init()
  - isFull()
  - isEmpty()
  - push(value)
  - pop()

Stacks: Example

- A fun exercise

Queues

- A queue is similar to the other data structures we've covered
  - Stores collection of elements
  - Restricts which element may be accessed
- Unlike stacks, queues are FIFO: First In First Out
- Think of the waiting queue at the DMV
  - Get a ticket
  - First ticket gets served first

Queues

- Words are added at the *tail*
- Words are removed from the *head*
- The position of the *tail* and *head* move as elements are added.

Queues: Array or Linked List?

- How do each need to operate?

Queues: ADT

Queue:

- private:
  - head
  - tail
  - data
  - queueSize
  - maxQueue
  - isEmpty()
  - isFull()
- public:
  - Init()
  - enqueue(value)
  - dequeue()