Data Structures

Arrays:

- Collection of same data type objects
- Allocates memory in a contiguous fashion

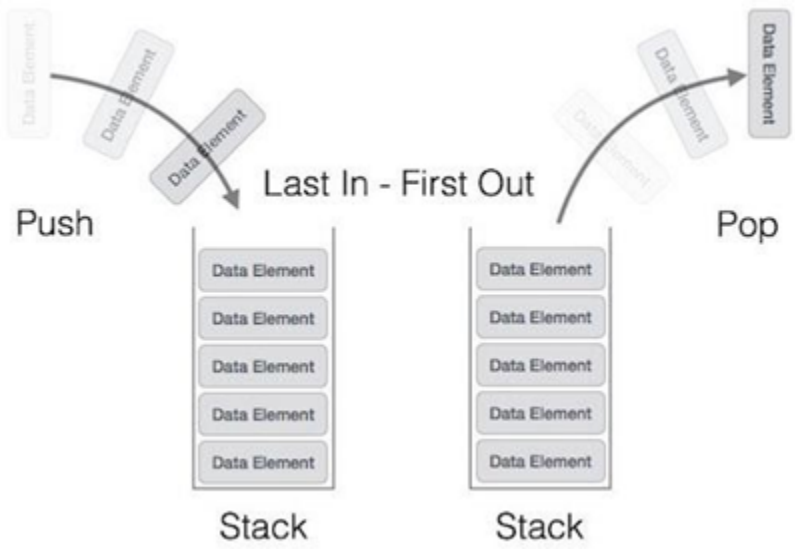
Vectors:

- Array like data types that can expand and shrink
- Can access any element anywhere
- push_back(element) to add element

Linked Lists:

- Linear data structure in which elements are not stored at contiguous memory locations
- Consists of a collection of nodes
    - Each node contains 2 main things, pointers to other nodes and data
    - Traditionally implemented as structs
    - But could be used as a class
        - Specify public access
        - Provide accessor/mutator methods
        - Make the linked-list class a friend of the node class
- Basic Operations:
    - Insertion - adds an element to the list
    - Deletion - deletes an element from the list
    - Display - displays the list
    - Search - searches an element using the given key
    - Delete - deletes an element using the given key
- Pointer called head/front that points to the first node
- Last node points to null ptr
- Insertion and deletion
    - A pointer that moves down a linked list (iterator) is used to insert or delete a node
    - Run time for a linked-list method is usually linear
    - The key to insertion and deletion methods in a linked list is to move the iterator to the node before the change
    - Often have to write code that divides into two cases depending on whether the head pointer needs to change or not
- Doubly linked lists
    - Insertion and deletion now have 4 possible cases
    - Neither first and last pointer change
    - First pointer needs to change
    - Last pointer needs to change
    - Insertion when empty or deletion of only node:
        - Both first and last pointer change
    - To reduce insertion and deletion to a single case, have dummy nodes for the first and last node
        - Head and tail

Stacks:

- ADT
- LIFO (last in first out)
- Basic operations
    - Insert = push (inserts value on top)
    - Remove = pop (removes value at top
    - 

Queues:

- ADT
- Open at both ends
- Basic operations
    - Insert = enqueue - the end that is always used to insert data (inserts to back)
    - Remove = dequeue - the end that is used to remove data (removes in front)
- FIFO (first in first out)

Priority Queues:

- Type of queue in which each element is associated with priority value
    - Elements are served on the basis of their priority
- Element with highest value is highest priority