Notes

Chapter 3: Stacks and Queues

Stacks

- A stack uses LIFO (last-in first-out) ordering
- uses the following operations
  - Pop(): Remove the top item from the stack
  - Push(item): Add an item to the top of the stack
  - Peek(): Return the top of the stack
  - isEmpty(): Return true iff the stack is empty
- Unline an array, a stack does not offer constant-time access to the ith item. However, it does allow constant time adds and removes, as it doesn't require shifting elements around.
- One case where stacks are often is in recursive algorithms. Sometime you need to push temporary data onto a stack as you recurse, but the remove them as you backtrack.
- A stack can also be used to implement a recursive algorithm iteratively.

Queues

- A queue implements FIFO (first-in, first-out) ordering. Like a line in a ticket stand.
- It uses operations
  - add (item): Add an item to the end of the list. (enqueue)
  - remove (): Remove the first item in the list
  - peek(): Return the top of the queue
  - isEmpty(): Return true iff the queue is empty
- A queue can also be implemented with a linked list. In fact, they are essentially the same thing, as long as items are added and removed from opposite sides.
- One place where queues are often used is in breadth-first search or in implementing a cache.