Stack Queue Deque

- Fibonacci Fun
  - repeated squaring
- ADT vs Data Structures
- Queues
- Stacks
- Deques
  - implementations with arrays, circular arrays, linked lists

- Linked lists Doubly-linked lists Skip lists
  - working with pointers

Complexity Analysis

- Time and Space complexity
- Algorithm Analysis: Counting steps
- Asymptotic Notation

$$O \qquad \Omega \qquad \Theta$$

If $f(n) \in O(g(n))$ then $2^{f(n)} \in O(2^{g(n)})$. True/False?

- Runtime Examples
  $lg(n!) \in \Theta(?)$
- Problem Complexity
  What is the time complexity of sorting by comparisons?

Priority Queues

- Rooted Trees, Briefly
- Priority Queue ADT
- Heaps
  - nearly complete binary tree
  - nifty representation as an array

Is

2	7	3	10	8	4	12

a min-heap?

  - Implementing Priority Queue ADT
    - SwapDown
    - SwapUp
    - Heapify
  - Analysis of Heapify
  - Brief introduction to d-Heaps

Induction

- Thinking Recursively
- Recursion Examples
- Analyzing Recursion: Induction and Recurrences
  - Use induction to prove correctness of recursive algorithm
  - Forming recurrences expressing running time
  - Solving recurrences (substitution method) and induction
  - Analysis of running time using a recursion tree
- Analyzing Iteration: Loop Invariants
  - Proof by induction on the number of iterations!
- How Computers Handle Recursion
  - Recursion and the Call Stack
  - Iteration and Explicit Stacks
  - Tail Recursion
  - Removing tail recursion

Sorting

- Comparing Sorting Algorithms
  - worst-, best-, average- case running times
  - stable sort
  - in-place
- Insertion Sort
- Heapsort
- Mergesort
- Quicksort
  How long would Quicksort run if it included the pivot in the second partition (in the worst case)?
- Complexity of Sorting
  - decision tree

Hashing

- Constant-Time Dictionaries?
- Hash Tables

  Are hash tables good for range queries?
- Hash Functions
- Collisions and the Pigeonhole Principle
  - load factor
- Collision Resolution:
  - Separate Chaining
  - Open Addressing

Search Trees

- Binary Trees
- Binary Search Trees
- Insertion, Deletion
  - reference parameters
- pre/in/post-order traversal
  Write code to reverse a binary search tree.
- Some troubling questions

AVL trees

- Balance implies shallow (shallow is good)
- How to achieve balance
- Single and double rotations
  What is the result of `rotateLeft(x); rotateRight(x);`?
- AVL tree implementation

B^+-Trees

- Minimizing disk I/Os
- B^+-Tree properties
  - M and L parameters
- Implementing B^+-Tree insert and delete

Parallelism

- History and Motivation
- Parallelism versus Concurrency
- Counting Matches in Parallel
- Divide and Conquer
- Fork and Join
- Reduce and Map
- Analyzing Parallel Programs
  - DAG
  - Work and Span
  - Ahmdahl's Law
- Parallel Prefix Sum

Graphs

- Topological Sort: Sorting vertices
- Graph ADT and Graph Representations
- Graph Terminology
- Graph Algorithms
  - Depth-First Search Breadth-First Search
  - Shortest Path (Dijkstra's Algorithm)
  - Minimum Spanning Tree (Kruskal's Algorithm)