Final Review

Pointers

- Referencing with &
- Dereferencing with *
- Pointer Independence
- Assigning through *
- Pointers as Variables

Linked List

- The `node` class
- Iterating through a Linked List
- Making a Linked List
- Operations (add to head, remove from head, etc)

Stacks

- LIFO
- Top, Push, Pop
- Depth First Search
- Applications

Queues

- FIFO
- Front, Push (enqueue), Pop (dequeue)
- Breadth First Search
- Applications

Analysis of Algorithms

- Selection Sort
- Insertion Sort
- Fisher-Yates
- Binary Search
- Merge Sort

Analysis of Algorithms

- Big O Simplification
- Dominance Relations
- Arithmetic Series $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$
- Analysis of simple functions (iterative and recursive)

Recursion

- What is recursion?
- Basics
- Backtracking
- Minimax

Types of Questions

- multiple choice/conceptual
- multiple choice/find the bug
- what does this code do/print?
- analysis of simple functions (as on worksheet W10)
- coding - could require the use or manipulation of:
  -- linked lists
  -- stacks
  -- queues
  -- simple recursion (no backtracking, minimax)

Sets

- Holds *unique* elements
- Ordered Set
- Unordered Set
- Iterate via *Iterators*
- Efficient at **find, insert, remove**

Maps

- Associating *keys* with *values*
- Keys must be *unique;* values may be anything
- Ordered / Unordered Maps
- Efficient at getting a value given a key, putting a key/value pair, remove key/value pair, update value given key, and determine if map has a key
- Pair class
- Difference between .insert(), .emplace(), []
- Editing values without making copies

Hashtables

- O(1) table lookups
- Basic idea: convert key to hash code, find index, store key at index
- Collisions (and chaining)
- What constitutes a `good' hash function?
- What data structures use hashtables?

Dynamic Allocation of Memory

- Array variables are pointers
- Pointer arithmetic
- Array limitations conquered by Dynamic Array Allocation
- Where does memory come from?
- Difference between creating new objects in Stack vs Heap
- Dynamic Memory Don'ts
- Deleting Dynamically Allocated Memory

Operator Overloading

- Member vs Non-member functions
- Mixed Overload
- How to overload

Big 3

- Copy Constructor
- Assignment Operator
- Destructor
- Shallow vs Deep Copy
- Default behaviors and how to fix them

Templates

- Purpose of templates
- Function vs Class Templates
- How to apply templating

Binary Trees

- Is empty, or a root node with a left child and a right child, each of which is a binary tree
- Pre, in, and post order traversals
- Count number of nodes

Binary Search Trees

- Data structure for holding comparable elements
- Underlying structure for sets, maps (BSTs)
- Nodes hold unique data values and pointers to child nodes
- Search, insert, remove operations and their complexity
- Understanding of self-balancing trees

Inheritance

- Inheritance serves various functions
  - Modeling of class relationships
  - Code reuse
  - Subtyping/polymorphism
- Override
- Virtual and Polymorphism
- Pure virtual and abstract classes