EECS 330 Practice Questions:

Questions From Midterm 1:

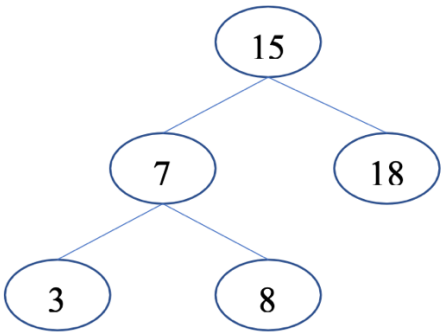
1. Explain Advantage of implementing iterator class for the vector and linked list data structures
2. Given a Linked List P and a vector L, both containing integers. We want to print the integers in P by taking the integers in L as the indexes. Ex: when L = (1, 4, 6) we want to print P[1], P[4], P[6]. Discuss your print function complexity when:
    a. P is singly-linked and L is sorted
    b. P is singly-linked and L is unsorted
    c. P is doubly-linked and L is sorted
    d. P is doubly-linked and L is unsorted
3. Mary has a little lamb. She loves the little lamb so much that whenever it is a sunny day and PetSmart is open, she will take the little lamb to PetSmart for fur care. We know that PetSmart opens on Mondays, and this Monday was sunny. Unfortunately, the bus route between Mary's home and PetSmart was not running this Monday because of a Union Strike. Do you think Mary took her Lamb to PetSmart this Monday?
    a. Write your proof in Symbolic Form
    b. Write your proof in Plain English
4. Given two linked lists L_1 and L_2, both contain integers. Devise an algorithm to compute $L_1 \cup L_2$. Pseudocode is sufficient. What is the time complexity of your algorithm?
5. Describe the advantages of passing rvalue reference as parameter
6. What is the worst-case scenario time complexity for inserting a data element. Into an unsorted array?
7. Implement the "reserve()" function in the vector class

    - Hint: you have the following definition in the vector class:
        - int *data;
        - int theSize;
        - int theCapacity;
        - void reserve(int newCapacity);
8. What are the "Big Five" defined in the context of C++ object interface?
9. Consider a doubly linked list designed for storing integer values. In a 64-bit machine, how much memory space does a regular (non-head, non-tail) node of the linked list occupy?

Questions from Past Finals (excluding questions with topics we haven't covered):

1. Describe two essential requirements for hash function design
2. Recall AVL tree,
    a. What is the AVL Property

3. Draw the resulted AVL tree after inserting 10 into the following AVL tree



4. Give the pseudocode of the Dijkstra's algorithm for finding the shortest path from an edge-weighted graph that contains no negative edge. Let the source node be *s,* and your algorithm should find the shortest path between *s* and all other nodes in the graph
5. Prove that Dijkstra's algorithm is correct when assuming no negative edge. And give a counterexample that the algorithm will not work if the graph contains negative edges
6. Write the pseudocode for depth-first search (DFS) and breadth-first shirt (BFS) algorithms. Let s be the source node. Your algorithm should print out the lists of visited nodes
7. Write the C++ code for the top(), enqueue(), dequeue() operations and the destructor function of a queue data structure. The data structure is implemented using singly-linked lists.
    a. Also implement if the data structure is implemented with vector (shouldn't make much of a difference)
8. Show that all comparison-based sorting algorithms have a time complexity lower bound of O(nlogn), where n is the number of items to be sorted. You can assume each comparison takes O(1) time.
9. Show the time complexity of the quicksort algorithm?
10. Describe the heap property and implementation of the priority queue. Show that you can perform enqueue() and dequeue() in O(logn) time, where n is the number of elements in the priority queue
11. Describe the implementation of the binomial queue. Show that you can perform merge() in O(logn) time where n is the total number of elements in both the binomial queues
12. In the context of C++ STL, explain what are the "sequential container", "associative container", and "container adapter". Given two examples for each of the three categories

<u>Noteworthy functions to be able to Implement:</u>

<u>Vector:</u>

<u>Linked-List:</u>

Infix Calculator:

Hash Table:

Binary Search Tree:

Binary Heap: