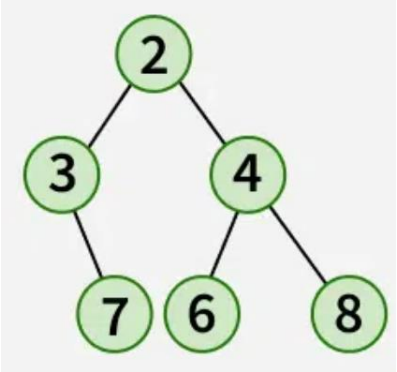
COMP 241Review

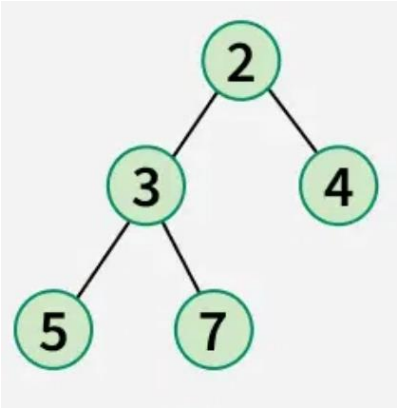
Topics:

- This exam is cumulative and covers previous concepts like Java programming, the n^2 sorting algorithms, abstract data types, interface vs implementation, and especially big O analysis. Arrays, linkedLists, stacks, queues, trees, hashtables, sets, and maps are also covered by the exam.
- Graphs
- Heaps
- Priority queues
- Quicksort, merge sort, and heap sort
- Dijkstra's Algorithm
- Breadth-first search, depth first search

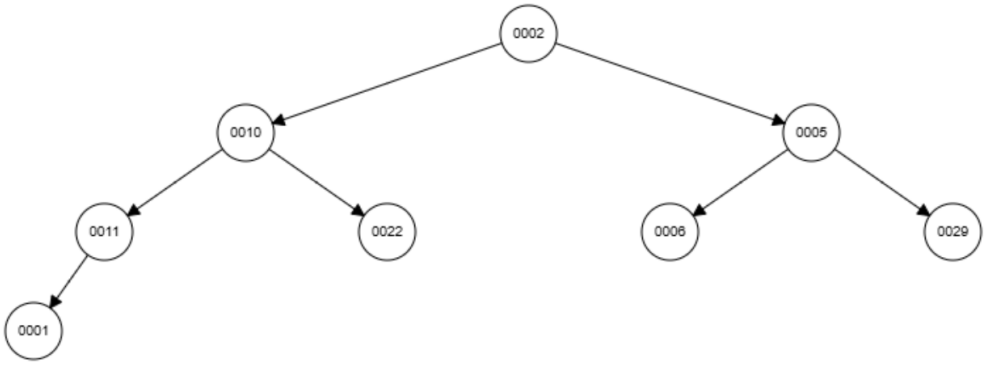
1. Which of the below are valid min heaps?



Invalid! The bottom row is not left-aligned.



Valid! It follows the min rule and is a balanced tree.



Invalid! The 1 on the bottom-left contradicts the min-heap rule, since it is smaller than its parent (11).

- For each of the heaps that are incorrect, why are they not a heap?

- What steps would you follow to apply heapsort to any of the heaps above? **Since they already are heaps, we would need to remove the top element (the smallest) n times and store each removed element in a list/array.**

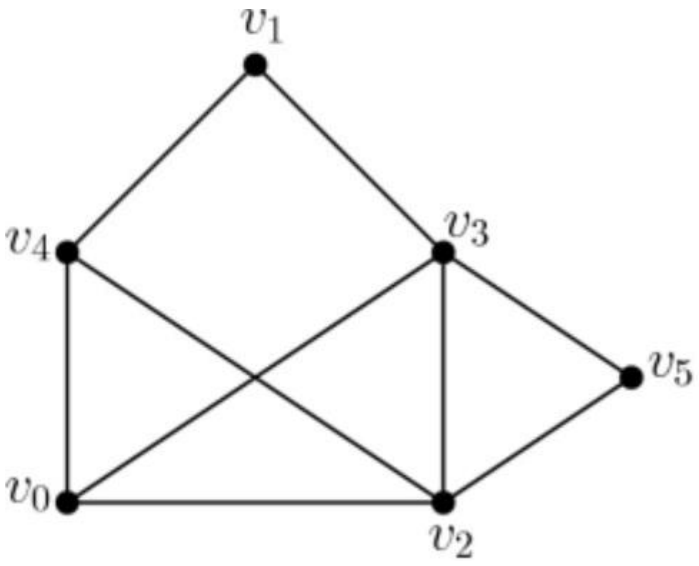
- For the last heap pictured here, if this heap was stored in an array, what element would index 3 be? (That is, array[3].)

Assuming the heap stayed as-is (an invalid heap!), array[3] would be 0011.
If the heap was first re-heaped, then array[3] would be 0010.

2. For the graph below, in what order would the vertices be visited in a breadth-first search? What about a depth-first search? The starting position is v0, and lower numbered nodes should be expanded first in case of a tie.

   BFS: V2, V3, V4, V5, V1

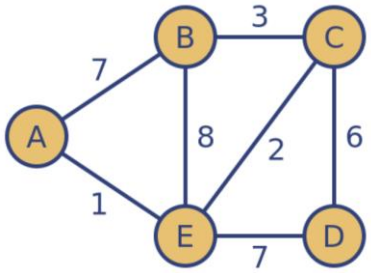
   DFS: V2, V3, V1, V4, V5



What is the big O running time of this algorithm?

O(edges + nodes)…For a strongly connected graph, that's equivalent to a worst-case O(n^2)

3. Using Dijkstra's Algorithm, find the cheapest path between D and B. Show your work!



Nodes visited: D(0), C(6), E(7), A(8), B(9)
Shortest path: D-C-B

4. Given an input of 4, 3, 1, 5, 8, 11, 2, 7, show the steps to sort this input using merge sort. Next, show the steps to sort the same array using quicksort.

4, 3, 1, 5, 8, 11, 2, 7

Mergesort:

Split the problem into two: 4 3 1 5 and 8 11 2 7

Split: 4 3 ; 1 5 ; 8 11 ; 2 7

Split into single digits

Combine! 3 4 ; 1 5 ; 8 11 ; 2 7

Combine: 1 3 4 5 ; 2 7 8 11

Combine: 1 2 3 4 5 7 8 11

Quicksort (using first element as pivot):

First pivot is 4

Numbers:

3, 1, 2 4 5, 8, 11, 7

Left side:

Pivot is 3; 1, 2 3

Pivot is 1; 1 2

Right side:

Pivot is 5 5 8, 11, 7

Pivot is 8 7 8 11

Now that all pivots have been selected, go back through the results, and your solution should be complete!

1, 2, 3, 4, 5, 7, 8, 11