Midterm Review

Familiarize yourself with the Core Objectives for each lesson (these accompany the lecture slides for each lesson).

1. General ideas of how BubbleSort, SelectionSort, InsertionSort work and running time.
2. Know general ideas of how LibrarySort refines InsertionSort and running time for LibrarySort.
3. Know the MergeSort and QuickSort algorithms. Be able to explain worst case, best case, average case and running time in each case.
4. Know different strategies of picking the pivot for QuickSort and compare the performance of them.
5. Know how to determine whether a sorting algorithm is stable or not.
6. Know the definition of *inversion-bound sorting algorithm,* the fact that on average, inversion-bound algorithms run in $\Omega(n^2)$, and the reason that this bound is valid.
7. Know the lower bound theorem for comparison-based sorting algorithms. In particular, be able to use the result that every comparison based sorting algorithm, running on an input array of size n, requires at least $\lceil \log(n!) \rceil$ comparisons in the worst case.
8. Know the definitions of Big-O and its relatives. Be able to determine when a function belongs to one of these complexity classes in simple cases. You will be able to use the limit definition.
9. Be familiar with BucketSort and RadixSort – be able to carry out the steps to solve a sorting problem and to give a running time analysis.
10. Be familiar with the three ways to determine the running time of a recursive algorithm (Guessing Method, counting self-calls, Master Formula). And decide when to use what.
11. Be familiar with Binary Search algorithm.
12. Be familiar with pseudocode and write algorithms in pseudocode. (Minor details will not matter.)
13. Know how to prove correctness of an iterative algorithm and a recursive algorithm.
14. Know average case running times for the primary operations on lists, stacks, queues, and hashtables and the worst cases for each data structure.
15. *Defintions* Be familiar with the definitions of the following concepts

- Heap(MinHeap and MaxHeap)
- red-black tree
- AVL tree

16. Be able to answer questions about BSTs:
- For red-black trees, know the result that every red-black tree has height $\leq 2\log(n+1)$
- Simple reasoning about red-black trees (like: can there be a red-black tree with 4 vertices, all black?)

17. Be able to do an insertion sequence for a red-black tree and (possibly) the steps for doing sorting (usual BST style of sorting).
18. Be able to work with a heap and to carry out the steps (Phase I and Phase II) of HeapSort.
19. Know the BottomUpHeap algorithms, both the recursive and iterative versions.
20. Heaps:
    a. Know that the running time of insertion, deletion and search in a heap is O(log n)
    b. Know that for a completedly filled binary tree/heap with height h, n nodes

$$n = 2^{h+1}-1$$