Midterm Review

Some notes:
Exam hours: 10:00 – 12:15 (Come at least 10 minutes earlier)
Classroom arrangement: VH 32 [10-8882, 61-0554] VH 29 [61-0555, 98-6814]
Calculus Reference and Master Formula will be provided in the exam.
No blank paper
No restroom break
No access to laptops, internet, phones, books, or notes

Important points you need to know for the exam:

1. Know the 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.
2. General ideas of how BubbleSort, SelectionSort, InsertionSort work and running time.
3. Know general ideas of how LibrarySort refines InsertionSort and running time for LibrarySort.
4. Know the MergeSort and QuickSort algorithms. Be able to explain worst case, best case, and average case running time.
5. Know different strategies of picking the pivot for QuickSort.
6. Know the QuickSelect algorithm.
7. Know how to prove correctness of recursive algorithms.
8. Know how to determine whether a sorting algorithm is stable or not.
9. 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.
10. 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.
11. 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.
12. Be familiar with the two ways to determine the running time of a recursive algorithm (counting self-calls, Master Formula).
13. Be familiar with Binary Search algorithm and know the different ways to compute its running time.

14. Be familiar with pseudocode and write algorithms in pseudocode. (Minor details will not matter.)
15. Know general ideas of the data structures – lists(ArrayList and LinkedList), stacks, queues, hashtables and BSTs. And running times for their primary operations.
16. Be able to create red-black trees using insertion sequences and the steps for doing sorting (usual BST style of sorting). Know the running time of primary operations on red-black trees.