|  |  |  |
| --- | --- | --- |
| **Data Structures** | **Algorithms** | **Concepts** |
| Linked Lists | Breadth-First Search | Bit Manipulation |
| Tree, Tries, Graphs | Depth-First Search | Memory(Stack vs Heap) |
| Stacks & Queues | Binary Search | Recursion |
| Heaps | Merge Sort | Dynamic Programming |
| Vectors/ArrayLists | Quick Sort | Big O Time & Space |
| Hash Tables |  |  |

**Walking through a problem:**

Listen Carefully – recorded any unique information from the problem, recorded down question on the board

Draw an example – creating sample that should be specific using real numbers or strings

State a Brute Force – make a simple algorithm

Optimize – Look for any unused information, fresh sample, make time vs space tradeoff, sorting data for long run, hash table

Walk Through – pseudocode

Implement – coding in the far top left corner, avoid slant messy code, explain out loud what you like to test, use classes for other function, good variable name

Test – conceptual test for each line of code, look for spot that might cause problem

Pg 67