1-2 Weeks DSA Brush-Up PLAN

means Leetcode Problem Number

ARRAY

1. Kadane's Algorithm (maximum sum subarray)
2. N/K Repeat number (ex. N/3 on interviewbit: use stackoverflow)
3. 3 sum (#15)
4. Kth smallest/Largest number (using quickselect)
5. All sorting algorithms
6. All searching algorithms
7. Trapping Rainwater Problem (#42)

LINKED LIST

1. Reverse LL in blocks of size K (LL-> Linked List)
2. Reorder List (#143)
3. Merge 2 sorted lists
4. Cycle detection/removal in LL
5. Add 2 numbers given as linked list

STACK

1. Implement stack
2. Stock Span Problem
3. Implement Queue using stack
4. Valid Parenthesis Problem

QUEUE

1. Implement Queue
2. Implement Stack using Queue
3. Circular Tour Problem

STRING

1. KMP Pattern Matching algorithm
2. Rabin Karp Algorithm (Rolling Hash)
3. Reverse all words in a text
4. Prefix->Postfix, Postfix->Infix etc. conversions
5. Word break problem (Dynamic Programming)
6. Group Anagrams (#49)

HASH (MAP)

1. Implement Hash
2. 4 Sum (#18)
3. Longest Substring without repeating characters (#3)

HEAP

1. Heap Implementation
2. Build Heap
3. Heap sort
4. Merge K sorted Lists (#23)
5. Last Stone Weight (#1046)

2-D Matrix

1. Rotate Matrix by 90/180/270
2. Rotate matrix spiral rings like a rotating spiral lock

Segment Tree

1. Implement Segment Tree
2. Range Sum Query
3. Range Minimum Query
4. Implement Lazy Propagation in Segment Tree

Binary Indexed Tree

1. Implement BIT
2. The Skyline Problem (#218)

TRIE

1. Count number of strings with a given prefix
2. Largest duplicate substring (#1044)

Dynamic Programming

1. Count Subsets with Sum X (Knapsack)
2. Target Sum Problem (#494)
3. Coin Change Problem (#322)
4. Coin Change 2 (#518)
5. Longest Repeating Subsequence (LCS variant)
6. Russian Doll Envelope (LIS variant #354)
7. Minimum Edit Distance
8. Kadane's Algo (#918->Max sum circular subarray)
9. Unique Paths 2 (DP on GRID #63)
10. Matrix Chain Multiplication (Balloon Burst Problem #312)
11. Egg Dropping Problem (MCM variant)

GRAPH

1. DFS/BFS algorithm
2. Number of islands (#200)
3. Topological Sort (Course schedule 2 #210)
4. Tarjan's Algorithm (Critical Connections in a network #1192)