Day1: (Arrays)

1. **Find the duplicate in an array of N+1 integers.**
2. **Sort an array of 0's 1's 2's without using extra space or sorting algo**
3. **Repeat and Missing Number**
4. **Merge two sorted Arrays without extra space**
5. **Kadane's Algorithm**
6. **Merge Overlapping Subintervals**

Day2: (Arrays)

1. **Set Matrix Zeros**
2. **Pascal Triangle**
3. **Next Permutation**
4. **Inversion of Array (Using Merge Sort)**
5. **Stock Buy and Sell**
6. **Rotate Matrix**

Day3: (Arrays/maths)

1. **Search in a 2D matrix**
2. **Pow(X,n)**
3. **Majority Element (>N/2 times)**
4. **Majority Element (>N/3 times)**
5. **Grid Unique Paths**
6. **Reverse Pairs (Leetcode)**
7. **Go through Puzzles from GFG (Search on own)**

Day4: (Hashing)

1. **2 Sum problem**
2. **4 Sum problem**
3. **Longest Consecutive Sequence**
4. **Largest Subarray with 0 sum**
5. **Count number of subarrays with given XOR(this clears a lot of problems)**
6. **Longest substring without repeat**

Day5: (LinkedList)

1. **Reverse a LinkedList**
2. **Find middle of LinkedList**
3. **Merge two sorted Linked List**
4. **Remove N-th node from back of LinkedList**
5. **Delete a given Node when a node is given. (0(1) solution)**

6. **Add two numbers as LinkedList**

Day6:

1. **Find intersection point of Y LinkedList**
2. **Detect a cycle in Linked List**
3. **Reverse a LinkedList in groups of size k.**
4. **Check if a LinkedList is palindrome or not.**
5. **Find the starting point of the Loop of LinkedList**
6. **Flattening of a LinkedList**
7. **Rotate a LinkedList**

Day7: (2-pointer)

1. **Clone a Linked List with random and next pointer**
2. **3 sum**
3. **Trapping rainwater**
4. **Remove Duplicate from Sorted array**
5. **Max continuous number of 1's**

Day8: (Greedy)

1. **N meeting in one room**
2. **Activity Selection**
3. **Greedy algorithm to find minimum number of coins**
4. **Fractional Knapsack Problem**
5. **Minimum number of platforms required for a railway**
6. **Job sequencing Problem**

Day9: (Backtracking)

1. **N queens Problem**
2. **Sudoku**
3. **M coloring Problem (Graph prob)**
4. **Rat in a Maze**
5. **Print all Permutations of a string/array**
6. **Word Break (print all ways)**

Day10:

1. **Combination sum-1**
2. **Combination sum-2**
3. **Palindrome Partitioning**

4. **Subset Sum-1**
5. **Subset Sum-2**
6. **K-th permutation Sequence**

Day11: (Divide and Conquer)

1. **1/N-th root of an integer (use binary search) (square root, cube root, ..)**
2. **Matrix Median**
3. **Find the element that appears once in sorted array, and rest element appears twice (Binary search)**
4. **Search element in a sorted and rotated array/ find pivot where it is rotated**
5. **Median of 2 sorted arrays**
6. **K-th element of two sorted arrays**

Day12: (Bits) (Optional, very rare topic in interviews, but if you have time left, someone might ask)

1. **Check if a number if a power of 2 or not in O(1)**
2. **Count total set bits**
3. **Divide Integers without / operator**
4. **Power Set (this is very important)**
5. **Find MSB in o(1)**
6. **Find square of a number without using multiplication or division operators.**

Day13: (Stack and Queue)

1. **Implement Stack / Implement Queue**
2. **BFS**
3. **Implement Stack using Queue**
4. **Implement Queue using Stack**
5. **Check for balanced parentheses**
6. **Next Greater Element**

Day14:

1. **Next Smaller Element**
2. **LRU cache (vvvv. imp)**
3. **Largest rectangle in histogram**
4. **Sliding Window maximum**
5. **Implement Min Stack**
6. **Rotten Orange (Using BFS)**

Day15: (String)

1. **Reverse Words in a String**

2. **Longest Palindrome in a string**
3. **Roman Number to Integer and vice versa**
4. **Implement ATOI/STRSTR**
5. **Longest Common Prefix**
6. **Rabin Karp**

Day16: (String)

1. **Prefix Function/Z-Function**
2. **KMP algo**
3. **Minimum characters needed to be inserted in the beginning to make it palindromic.**
4. **Check for Anagrams**
5. **Count and Say**
6. **Compare version numbers**

Day17: (Binary Tree) - Done

1. **Inorder Traversal (with recursion and without recursion)**
2. **Preorder Traversal (with recursion and without recursion)**
3. **Postorder Traversal (with recursion and without recursion)**
4. **LeftView Of Binary Tree**
5. **Bottom View of Binary Tree**
6. **Top View of Binary Tree**
7. **Right View of Binary Tree**

Day18: (Binary Tree) -Done

1. **Level order Traversal / Level order traversal in spiral form**
2. **Height of a Binary Tree**
3. **The diameter of Binary Tree**
4. **Check if Binary tree is height balanced or not**
5. **LCA in Binary Tree**
6. **Check if two trees are identical or not**
7. **Vertical order traversal.**

Day 19: (Binary Tree) -Done

1. **Maximum path sum**
2. **Construct Binary Tree from inorder and preorder**
3. **Construct Binary Tree from Inorder and Postorder**
4. **Symmetric Binary Tree**
5. **Flatten Binary Tree to LinkedList**
6. **Check if Binary Tree is mirror of itself or not**

Day 20: (Binary Search Tree)-Doing

1. **Populate Next Right pointers of Tree**
2. **Search given Key in BST**
3. **Construct BST from given keys.**
4. **Check is a BT is BST or not**
5. **Find LCA of two nodes in BST**
6. **Find the inorder predecessor/successor of a given Key in BST.**

Day21: (BinarySearchTree)

1. **Floor and Ceil in a BST**
2. **Find K-th smallest and K-th largest element in BST (2 different Questions)**
3. **Find a pair with a given sum in BST**
4. **BST iterator**
5. **Size of the largest BST in a Binary Tree**
6. **Serialize and deserialize Binary Tree**

Day22: (Mixed Questions)

1. **Binary Tree to Double Linked List**
2. **Find median in a stream of running integers.**
3. **K-th largest element in a stream.**
4. **Distinct numbers in Window.**
5. **K-th largest element in an unsorted array.**
6. **Flood-fill Algorithm**

Day23: (Graph)

1. **Clone a graph (Not that easy as it looks)**
2. **DFS**
3. **BFS**
4. **Detect A cycle in Undirected Graph/Directed Graph**
5. **Topo Sort**
6. **Number of islands (Do in Grid and Graph both)**
7. **Bipartite Check**

Day24: (Graph)

1. **SCC(using KosaRaju's algo)**
2. **Djisktra's Algorithm**
3. **Bellman Ford Algo**
4. **Floyd Warshall Algorithm**
5. **MST using Prim's Algo**
6. **MST using Kruskal's Algo**

Day25: (Dynamic Programming)

1. **Max Product Subarray**
2. **Longest Increasing Subsequence**
3. **Longest Common Subsequence**
4. **0-1 Knapsack**
5. **Edit Distance**
6. **Maximum sum increasing subsequence**
7. **Matrix Chain Multiplication**

Day26: (DP)

1. **Maximum sum path in matrix, (count paths, and similar type do, also backtrack to find the maximum path)**
2. **Coin change**
3. **Subset Sum**
4. **Rod Cutting**
5. **Egg Dropping**
6. **Word Break**
7. **Palindrome Partitioning (MCM Variation)**

Day27:

1. **Revise OS notes that you would have made during your sem**
2. **If not made notes, spend 2 or 3 days and make notes from Knowledge Gate.**

Day28:

1. **Revise DBMS notes that you would have made during your semesters.**
2. **If not made notes, spend 2 or 3 days and make notes from Knowledge Gate.**

Day29:

1. **Revise CN notes, that you would have made during your sem.**
2. **If not made notes, spend 2 or 3 days and make notes from Knowledge Gate.**

Day30:

1. **Make a note of how will your represent your projects, and prepare all questions related to tech which you have used in your projects. Prepare a note which you can say for 3-10 minutes when he asks you that say something about the project.**

Hurrah!! You are ready for your placement after a month of hard-work without a cheat day.