1. https://www.hackerrank.com/challenges/johnland/problem

2.https://media.discordapp.net/attachments/903142711891202088/904379062976385024/IMG-20211031-WA0067.jpeg

3. https://leetcode.com/discuss/interview-question/1528662/determine-the-number-of-pairsijsuch-that-aiaj-and-ij-is-divisible-by-x

4. https://leetcode.com/discuss/interview-question/725801/amazon-shared-interest-problem

5. https://leetcode.com/discuss/interview-question/1509054/swiggy-oa-sde-1

6. https://www.interviewbit.com/problems/kth-manhattan-distance-neighbourhood/

7. mincost to travel with atmost k stops leetcode 787

8. leetcode 1504 gem question

9. All Topological Sorts of a Directed Acyclic Graph

10. implement LFU cache

11. https://www.hackerrank.com/challenges/manasa-loves-maths/problem

12. SQL query to find the name of the employee with kth largest salary

13. pairs with given sum bst

14. Detect cycle in graph

15. Check if large number is divisible by 8 - gfg
16. Vount vowels permutation - 1220 leetcode
17. Oops diamond problem (Java)
18. Design implement special stack all operation O1 – gfg
19. Wiggle sort
20. Delete all even node in linked list – gfg
21. Length of shortest chain to reach a target word – gfg
22. Count max points on same line – gfg
23. Non negative no. Missing in sorted array
24. Max path sum binary tree
25. Indexing compressing and chunking techniques (memory managemnet) (DBMS)
26. Timsort
27. Complexity of all sorting algos
28. Reason for use of OOPs and its paradigm (OOPs)
29. Number of triangles possible given n points in 2 arrays one having x another y
30. Abstraction and encapsulation (explain it to 7th grade student)😂
31. Dbms - indexing , dense vs sparse indexes, joins, types of joins, def. Of inner join.

Query for inner join

ACID props of dbms

32. Http and https diff
33. Explain kmp algo interview
34. Interview paging and difference between process and thread
35. Beautiful arrangement 1 and 2 leetcode 667 and 526
36. Delete node return forest leetcode 1110
37. Maximal rectangle leetcode 85
38. Rolling hash rabin karp
39. Regex matching
40. Activity selection greedy approach
41. Cherry pickup leetcode 741 dp
42. Is binary tree a bst using preorder
43. Leetcode943
44. Shortest superstring
45. Are trees similar
46. Largest pallindromic substring\
47. Dice sum target
48. Min taps to water a garden leetcode 1336

49. Infix to postfix

50. Min adjacent swap to make string pallindrome

51. https://leetcode.com/list/5npgbhnt

My seniors suggestion-

PLACEMENT PREPARATION SUGGESTED TIMELINE

● Choose a language (C++ / Java).

● Learn the syntax of the language.

● Learn stl/c++ thoroughly.

● If you choose java, be prepared with collections, comparator, Math class, Fast I/O.

● Be comfortable with the basic implementation. (Give time to this)

● Always upsolve the unsolved problems at the end of the contest on the day of the contest itself.

● You should be able to solve Div2C problem regularly in the contests. (Around 50% of the time)

● If your english is a bit weak, or you feel you are not confident enough to speak. Start talking with your friends and family in english.

● Try to cover these topics as much as you can: ○ Backtracking ○ Binary search ○ DP ○ Greedy ○ Graph ■ DSU ■ BFS/DFS related topics ■ Shortest path algorithms ■ Topo sort ■ Bipartite checking/ cycles/ colouring (study after the above topics are covered) ○ Segment Tree/ Fenwick Tree ○ Digit DP/ DP on trees/ DP with bitmasking ○ String algorithms (After Intern)

● Cover remaining algorithms.

● Practice/ give contests rigorously on leetcode or interviewBit.
● Try to do at least 30-40 questions (majorly medium and hard) of each topic from leetcode.

● Read GFG articles regularly (data structures/ algorithms) and also try to implement some of these side by side on leetcode.

● Study SQL, practice on hackerrank.

● Read CS subjects ○ OOPS ○ OS ○ DBMS ○ Computer Networks

● Prepare for MCQs. (GFG/ placement docs)

● Coding practice/ gfg/ cs subjects

● Try and give contests regularly (at least one in two days whatever level you're at) If there aren't any hosted on the famous websites or if they are tough, search for a new one or host one yourself (sources given below in "Suggested Resources" or give virtual contests).

● Interview prep ○ Cover important DS algo interview questions from GFG. You can follow this. Regularly use Leetcode to practice too, as you are generally asked to code in your interviews. ○ Revise CS subjects. ○ Puzzles: Gfg (1-85) ○

Prepare resume. ■ Projects ■ Technical skills- Only mention those in your resume in which you are confident. ○ Communication skills are very important. ■ Try giving mock interviews to friends or seniors. ■ HR questions.

● For implementation ○ A2oj ladders/ Rating filter on codeforces. (start from div2A, should be comfortable at least upto Div2C later) ○ Be comfortable with contests (codeforces/ atcoder/ leetcode). Leetcode contests are more relevant to placement preparation. ○ Hackerrank.

● For data structures/ algorithms ○ Array: leetcode ○ Math: leetcode ○ String: leetcode ○ Greedy: leetcode, gfg, POTW ○ Stack: gfg, leetcode ○ Queue: gfg, leetcode ○ Binary search: leetcode, POTW ○ Sorting: leetcode, algos from gfg(Insertion Sort, Bubble Sort, Selection Sort, Merge Sort, Quick Sort, Heap Sort) ○ Priority queue (STL): Heap (Aditya Verma), leetcode ○ Sliding Window, 2 pointer: leetcode ○ Hashing: leetcode ○ LinkedList: gfg ○ Bit Manipulation: Hackerearth theory, leetcode ○ Backtracking + recursion: gfg, leetcode ○ Divide and Conquer: gfg, leetcode (focus on merge sort method) ○ DP (focus on recursion) (Aditya Verma from youtube)

■ Leetcode or InterviewBit + (GFG 56 ques on algorithm page) ■ Digit Dp (codeforces)

■ DP with Bitmask (after graph) (Codeforces and Kartik Arora Videos)

■ DP on trees (after graph) (GFG/Codeforces) ○ Graph

■ For theory: CP Algo, hackerearth, gfg (not good codes)

■ For practice: leetcode, InterviewBit, Codeforces, Hackerearth (not good solutions)

■ Also, do refer POTW ● BFS ● DFS ● DSU (with Path compression) ● Topological Sort ● Bipartite ● Cycle, coloring ● Shortest paths (Djisktra, Bellman Ford, Floyd Warshal, BFS) ● Minimum Spanning Tree ● Flood Fill Algorithm ● Bridges articulation ● Strongly Connected Components ● Biconnected Components (optional, rarely asked) ● Hamiltonian cycle + path (Dp with bitmasking) ● LCA (from here) (Binary Lifting) ● Max flow Min flow (optional, rarely asked) ○ Binary Trees: gfg, InterviewBit ○ Binary Search Tree: gfg, InterviewBit ○ Heap: read operations from gfg, but implement using priority_queue ○ String Algos: ■ Refer POTW ● KMP algorithm: CP Algorithms (most Important) ● Z algorithm: Cp Algorithms (Rarely used) ● Hashing: CP Algorithms (Rarely Used)

● Advanced DS: (asked rarely) ○ Trie (Hackerearth) ○ Segtree with Lazy Propagation (Hackerearth, CP Algo, POTW) ○ AVL Tree: gfg (asked a lot in mcq) ○ Red Black Tree : gfg ○ Fenwick Tree: hackerearth, CP algo, POTW ○ Suffix Tree

● CS Subject ○ OOPS (GFG or learncpp.com or Saurabh Shukla YT videos)(for java : javatpoint.com) ○ OS (Yellow pages, Gate Smashers) ○ DBMS (Gate smashers) ○ Networking (Gate Smashers)

● Alternative coding websites/hosting contests yourself ○ https://atcoder.jp/ is a good website having questions that are a hybrid of leetcode and Codeforces questions. You can't host your own contests here, but at https://kenkoooo.com/atcoder/#/table/

you can host a contest consisting of atcoder problems ○ Codeforces lets you host your own contest ○ https://binarysearch.com/ is full of Leetcode problems, you can host your own contest topic wise or company wise (Easy, Med, Hard problems are enough, Harder level problems are a bit over the top is what I found) Placement Gyaaan: link GFG Placement 100 Course: Here Important Leetcode links: Here some of the important algorithms yt link : link CP algorithms is a good source for slightly complex algorithms which may come directly as a problem (eg. Josephus Problem (ye to karke hi jaana xD)) :

https://cp-algorithms.com/

The more questions you solve on leetcode, the more are your chances in cracking a good company.

All the best