LINKED LIST ROADMAP (SORTED + PATTERNS + INDEXING)

EASY LEVEL (1–20)

---

1. Implement Singly Linked List
2. Implement Doubly Linked List
3. Implement Circular Linked List
4. Insert at head
5. Insert at tail
6. Insert at position
7. Delete head
8. Delete tail
9. Delete at position
10. Search in LL
11. Length of LL
12. Find middle of LL — Slow/Fast
13. Check palindrome — Reverse + compare
14. Count nodes in loop — Floyd
15. Detect loop — Floyd
16. Find start of loop — Floyd
17. Remove loop — Floyd
18. Reverse LL (iterative)
19. Reverse LL (recursive)
20. Check if LL is circular

MEDIUM LEVEL (21–45)

---

21. Remove duplicates (sorted)
22. Remove duplicates (unsorted)
23. Intersection of two LLs
24. Add 1 to LL number
25. Add two numbers represented by LL
26. Rotate LL by K

27. Left/Right rotate
28. Segregate even/odd nodes
29. Rearrange odd-even index nodes
30. Delete N nodes after M nodes
31. Move last to front
32. Swap nodes pairwise
33. Remove nodes w/ greater value on right
34. Next greater node in LL
35. Merge two sorted LL
36. Merge K sorted lists
37. Sort LL with merge sort
38. Sort LL with quicksort
39. Flatten multilevel LL
40. Flatten binary LL (next + bottom)
41. Reverse LL in groups of K
42. Reverse alternate K nodes
43. Reverse between left–right
44. Reverse doubly linked list
45. Happy number (Floyd logic)

HARD LEVEL (46–70)

---

46. Clone LL with random pointer
47. LRU Cache
48. LFU Cache
49. Design Twitter feed
50. Browser History design
51. Add two LLs (forward order)
52. Multiply two LLs
53. Subtract LL numbers
54. Divide LL numbers
55. Reorder LL (1,n,2,n-1)

56. Split circular LL to two halves
57. Check if LL is sorted
58. Delete nodes with greater on right
59. LRU variants
60. LFU variants
61. Remove nodes leaving only distinct
62. Partition LL around value
63. Random node in LL (reservoir sampling)
64. LL binary to number
65. Plus/minus one on number-LL
66. Polynomial addition using LL
67. Polynomial multiplication LL
68. Deep copy multilevel LL
69. Intersection of 3 LLs
70. Polygonal chain logic using LL

BONUS (71–75)

---

71. Detect intersection without modifying LL
72. Delete node using only pointer
73. Nth node from end
74. GCD of LL values
75. Frequency of value in LL