HASHING ROADMAP (SORTED + INDEXED + PATTERNS)

EASY LEVEL (1–20)

---

1. Implement Hash Table
2. Implement HashMap (chaining)
3. Implement HashSet
4. Count frequency of elements
5. First non■repeating element
6. Second most frequent element
7. Check if two arrays are equal
8. Check anagram
9. Count distinct elements
10. Union/Intersection of arrays
11. Find duplicates using HashSet
12. Two-sum
13. Subarray with sum 0
14. Subarray with sum K exists
15. Longest subarray sum K (positive)
16. Pair sum exists
17. All pairs with sum K
18. Largest subarray equal 0s & 1s
19. Count subarrays with sum K
20. Group anagrams

MEDIUM LEVEL (21–50)

---

21. Count subarrays XOR = K
22. Longest subarray XOR = K
23. Longest subarray equal letters/digits
24. Subarray divisible by K
25. Happy number (hash cycle)
26. Detect cycles using hashset

27. Rabin–Karp
28. Polynomial rolling hash
29. Double hashing
30. Longest duplicate substring
31. Rotation check using hashing
32. Count distinct substrings
33. Longest consecutive sequence
34. Players zero/one loss
35. Smallest subarray with distinct chars
36. Longest substring without repeat
37. Longest substring with K distinct
38. Isomorphic strings
39. Custom sort string
40. Sort characters by frequency
41. Top K frequent elements
42. Duplicate chars in string
43. Common elements in 3 arrays
44. First element with freq >1
45. Check permutation in string
46. Anagram substring check
47. First occurrence using hashing
48. Minimum window substring
49. Longest substring equal 0,1,2
50. Subarray modulo hashing

HARD LEVEL (51–80)

---

51. Word pattern match
52. Repeated substring pattern (hash)
53. Longest palindromic substring (hash)
54. Strings differ by one substring
55. All anagrams in string

56. Distinct substrings O(n)
57. Longest happy substring
58. Longest nice substring
59. Largest subarray equal counts
60. Pos/neg equal subarray
61. Clone graph using map
62. Clone tree using map
63. Bipartite check using map
64. Eventual safe states
65. LRU Cache
66. LFU Cache
67. Randomized Set O(1)
68. Randomized Collection
69. Design Twitter
70. Time-based key-value store
71. Underground system
72. Rearrange string (heap+map)
73. Reorganize string K apart
74. Task scheduler
75. Hard min window substring
76. String equivalence hashing
77. Z-algo + hashing hybrid
78. Subsequence check using hash
79. Unique substring length K
80. Plagiarism detection hashing

BONUS (81–85)

---

81. Custom hash functions in Java
82. Load factor & rehashing
83. HashMap vs LinkedHashMap vs TreeMap
84. Mutable key problem in HashMap

85. HashSet vs TreeSet vs LinkedHashSet