STRING QUESTIONS ROADMAP (SORTED BY DIFFICULTY + PATTERNS)

EASY LEVEL (1–25)

----

1. Reverse a string — Two-pointer
2. Palindrome check — Two-pointer
3. Count vowels & consonants — Linear scan
4. Toggle case — ASCII ops
5. Frequency of characters — HashMap
6. Remove duplicate characters — Set
7. Sort characters — Sorting
8. Largest odd number in a string — Scan
9. Reverse words — Two-pointer
10. Reverse every word — Reverse ops
11. Valid palindrome II — Two-pointer
12. Remove outermost parentheses — Stack
13. Max nesting depth — Stack
14. Balanced string split — Counter
15. Isomorphic strings — Hash mapping
16. Check anagram — Frequency array
17. Check string rotations — s1 in s2+s2
18. Subsequence check — Two-pointer
19. Count substrings — Math
20. Implement atoi() — Parsing
21. Roman to Integer — Hashing
22. Integer to Roman — Greedy
23. Implement strstr() brute — Naive
24. Longest common prefix — Compare
25. Group anagrams — Hash map

MEDIUM LEVEL (26–55)

----

26. Smallest window substring — Sliding window

27. Longest substring with char replacement — Sliding window

28. Longest substring without repeat — Sliding window

29. Min additions palindrome — Two-pointer

30. Rabin–Karp — Hashing

31. Z-function — Z-array

32. KMP / LPS — Pattern matching

33. strstr() KMP — KMP

34. Longest palindromic substring — Expand center

35. Shortest palindrome — KMP

36. Longest happy prefix — KMP

37. Count palindromic subsequences — DP

38. Count palindromic substrings — Expand center

39. Longest repeating substring — Binary search + hash

40. Longest common substring — DP

41. Word break I — DP

42. Word break II — Backtracking + DP

43. Sort char by frequency — HashMap + heap

44. Count and Say — String building

45. Sum of beauty of substrings — Contribution

46. Minimum bracket reversals — Stack

47. Valid parentheses — Stack

48. First non repeating char — Frequency

49. Remove adjacent duplicates — Stack

50. String compression — Two-pointer

51. Binary string prefix check — Hashing

52. Custom sort string — Frequency

53. Rotate string — Slicing

54. Reorganize string — Heap

55. First occurrence optimized — KMP/Z

HARD LEVEL (56–80)

----

56. LCS — DP

57. Edit distance — DP

58. Min insertions palindrome — DP

59. Min deletions palindrome — DP

60. Distinct subsequences — DP

61. Wildcard matching — DP

62. Regex matching — DP

63. Longest duplicate substring — Binary search + hash

64. Longest prefix also suffix — KMP

65. Find all anagrams in string — Sliding window

66. Implement Trie — Trie

67. Longest word in dictionary — Trie

68. Autocomplete system — Trie

69. Palindrome partitioning — Backtracking

70. Palindrome partitioning II — DP

71. Remove invalid parentheses — BFS

72. Decode string — Stack

73. Encode string shortest — DP

74. Lexicographically smallest string operations — Greedy

75. Longest substring K repeats — Sliding window

76. Longest chain of substrings — DP

77. Longest palindromic subsequence — DP

78. Scramble string — DP

79. Interleaving string — DP

80. Count distinct substrings — Suffix array/Trie

BONUS IMPORTANT JAVA STRING QUESTIONS

----

81. Immutable strings vs StringBuilder

82. Char array ↔ string conversion

83. Rearranging to form palindrome

84. Count words manually

85. Duplicate chars without map

86. Print all permutations

87. Lexicographically next string

88. Reverse string recursion

89. Permutation substring check

90. Manual ASCII upper/lowercase conversion