

ALGOLAB 150

Algorithm Patterns.

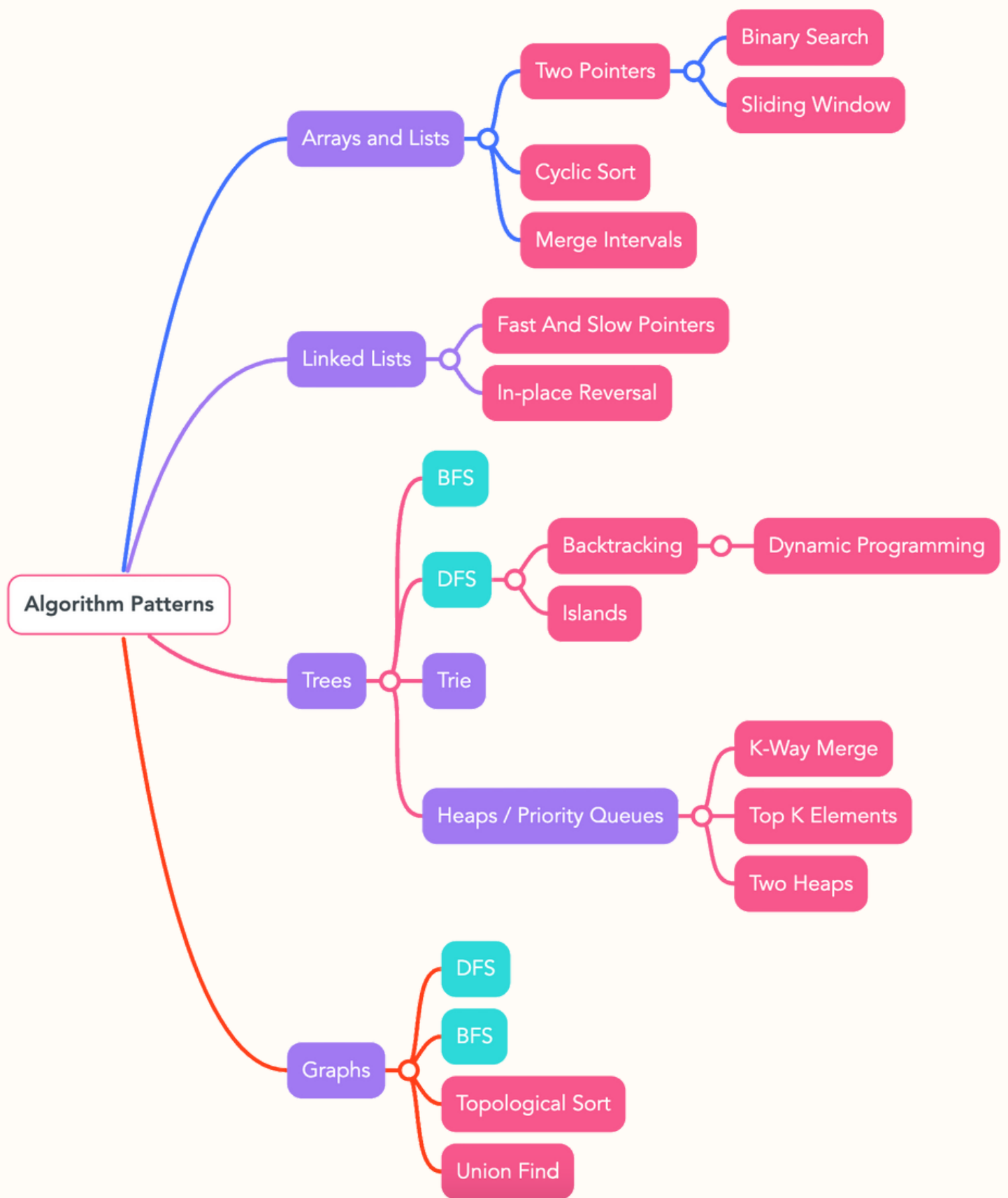
150 Problems.

Problem-Solving.

Ace Coding Interviews.



"KEY TO ACING
CODING INTERVIEWS"
— FAANG ENGINEER



Instructions

- Warm-up with easy problems first. Get few problems under your belt that you could solve under 15 minutes.
- Master one pattern at a time. Don't move forward until you become comfortable with current pattern.
- A problem can be solved with many different patterns, but to master the current pattern, solve the problem with the pattern it is listed under.
- Never spend more than one hour maximum per problem. If it is taking that much time, then this is not the right level for you yet. Go back to easier problems and redo them faster.
- If you cannot solve the problem within first 25 minutes, then get hints by looking at the solution from others.
- Repeat coding of each problem for three times. First time, it will take longer, but second and third time it will be very fast. With little extra time, you will gain a lot.
- After solving a problem, have a look at the solutions by others to pick good points and improve your solution.
- Join Algolab's video course to get video lessons, coding workspace, official solutions and solutions from others.

Warm Up

1. [Palindrome Number](#)
2. [Merge Two Sorted Lists](#)
3. [Single Number](#)
4. [Contains Duplicate](#)
5. [Valid Anagram](#)
6. [Fibonacci Number](#)
7. [Defanging An Ip Address](#)
8. [Running Sum Of 1d Array](#)
9. [Number Of Good Pairs](#)
10. [Richest Customer Wealth](#)
11. [Build Array From Permutation](#)
12. [Concatenation Of Array](#)
13. [Fizz Buzz](#)

Fixed Size Sliding Window

14. [Number Of Sub Arrays Of Size K And Average Greater Than Or Equal To Threshold](#)
15. [Substrings Of Size Three With Distinct Characters](#)
16. [Find All Anagrams In A String](#)
17. [Permutation In String](#)
18. [Maximum Average Subarray I](#)
19. [Diet Plan Performance](#)
20. [Find K Length Substrings With No Repeated Characters](#)
21. [Average Of All Subarrays](#)

Dynamic Size Sliding Window

- 22. [Longest Substring Without Repeating Characters](#)
- 23. [Max Consecutive Ones Iii](#)
- 24. [Max Consecutive Ones](#)
- 25. [Minimum Window Substring](#)

Binary Search

- 26. [Find In Mountain Array](#)
- 27. [Search In Rotated Sorted Array](#)
- 28. [Find First And Last Position Of Element In Sorted Array](#)
- 29. [Single Element In A Sorted Array](#)
- 30. [Binary Search](#)
- 31. [Find Smallest Letter Greater Than Target](#)
- 32. [Peak Index In A Mountain Array](#)

Two Pointers

- 33. [Container With Most Water](#)
- 34. [3sum](#)
- 35. [Two Sum If Input Array Is Sorted](#)
- 36. [Valid Palindrome](#)
- 37. [Remove Duplicates From Sorted Array](#)
- 38. [Remove Element](#)
- 39. [Move Zeroes](#)
- 40. [Reverse String](#)
- 41. [Sort Colors](#)
- 42. [Squares Of A Sorted Array](#)

Cyclic Sort

- 43. [Kth Missing Positive Number](#)
- 44. [First Missing Positive](#)
- 45. [Find All Duplicates In An Array](#)
- 46. [Find All Numbers Disappeared In An Array](#)
- 47. [Find The Duplicate Number](#)
- 48. [Set Mismatch](#)
- 49. [Missing Number](#)

Top K Element

- 50. [Least Number Of Unique Integers After K Removals](#)
- 51. [Sort Array By Increasing Frequency](#)
- 52. [Kth Largest Element In An Array](#)
- 53. [Sort Characters By Frequency](#)
- 54. [Top K Frequent Elements](#)
- 55. [Find K Closest Elements](#)
- 56. [K Closest Points To Origin](#)

Merge Intervals

- 57. [Non Overlapping Intervals](#)
- 58. [Merge Intervals](#)
- 59. [Insert Interval](#)
- 60. [Interval List Intersections](#)
- 61. [Employee Free Time](#)
- 62. [Meeting Rooms](#)
- 63. [Meeting Rooms li](#)
- 64. [Meeting Scheduler](#)

Linked List

- 65. [Reverse Linked List](#)
- 66. [Rotate List](#)
- 67. [Reverse Linked List II](#)
- 68. [Reverse Nodes In K Group](#)

Fast And Slow Pointers

- 69. [Linked List Cycle](#)
- 70. [Linked List Cycle II](#)
- 71. [Happy Number](#)
- 72. [Middle Of The Linked List](#)

Depth First Search (DFS)

- 73. [Path Sum Iii](#)
- 74. [Same Tree](#)
- 75. [Maximum Depth Of Binary Tree](#)
- 76. [Path Sum](#)
- 77. [Path Sum Ii](#)
- 78. [Binary Tree Maximum Path Sum](#)
- 79. [Sum Root To Leaf Numbers](#)
- 80. [Evaluate Boolean Binary Tree](#)
- 81. [Diameter Of Binary Tree](#)
- 82. [Range Sum Of Bst](#)
- 83. [Binary Tree Inorder Traversal](#)
- 84. [Find All The Lonely Nodes](#)

Breadth First Search (BFS)

- 85. [Binary Tree Level Order Traversal](#)
- 86. [Binary Tree Zigzag Level Order Traversal](#)
- 87. [Minimum Depth Of Binary Tree](#)
- 88. [Binary Tree Level Order Traversal II](#)
- 89. [N Ary Tree Level Order Traversal](#)
- 90. [Average Of Levels In Binary Tree](#)

Islands

- 91. [Flood Fill](#)
- 92. [Number Of Islands](#)
- 93. [Surrounded Regions](#)
- 94. [Walls And Gates](#)
- 95. [Find All Groups Of Farmland](#)
- 96. [Count Sub Islands](#)
- 97. [Minesweeper](#)

Topological Sort

- 98. [Course Schedule](#)
- 99. [Course Schedule II](#)
- 100. [Alien Dictionary](#)
- 101. [Minimum Height Trees](#)
- 102. [All Ancestors Of A Node In A Directed Acyclic Graph](#)
- 103. [Build A Matrix With Conditions](#)
- 104. [Find All Possible Recipes From Given Supplies](#)

Tries

- 105. [Word Break](#)
- 106. [Word Break II](#)
- 107. [Implement Trie Prefix Tree](#)
- 108. [Design Add And Search Words Data Structure](#)
- 109. [Word Search II](#)
- 110. [Concatenated Words](#)

Union Find

- 111. [Number Of Operations To Make Network Connected](#)
- 112. [Number Of Provinces](#)
- 113. [Remove Max Number Of Edges To Keep Graph Fully Traversable](#)
- 114. [Redundant Connection](#)
- 115. [Similar String Groups](#)
- 116. [Graph Valid Tree](#)
- 117. [Number Of Connected Components In An Undirected Graph](#)

Two Heaps

- 118. [Find Median From Data Stream](#)
- 119. [Sliding Window Median](#)
- 120. [IPO](#)
- 121. [Find Right Interval](#)

Backtracking

- 122. [Subsets](#)
- 123. [Subsets II](#)
- 124. [Permutations](#)
- 125. [Letter Combinations Of A Phone Number](#)
- 126. [Palindrome Partitioning](#)
- 127. [Combination Sum](#)
- 128. [Combination Sum II](#)
- 129. [Permutations II](#)

K Way Merge

- 130. [Merge K Sorted Lists](#)
- 131. [Kth Smallest Element In Sorted Matrix](#)
- 132. [Find K Pairs With Smallest Sum](#)
- 133. [Smallest Range Covering Elements From K Lists](#)
- 134. [Rearranging Strings K Distance Apart](#)
- 135. [Minimum Cost To Hire K Workers](#)

Dynamic Programming

- 136. [Fibonacci Number](#)
- 137. [Counting Bits](#)
- 138. [Is Subsequence](#)
- 139. [Get Maximum In Generated Array](#)
- 140. [Climbing Stairs](#)
- 141. [Minimum Cost Climbing Stairs](#)
- 142. [Pascal Triangle](#)
- 143. [Pascal Triangle II](#)
- 144. [Nth Tribonacci Number](#)
- 145. [Unique Paths](#)
- 146. [Unique Paths](#)
- 147. [All Possible Full Binary Trees](#)
- 148. [Minimum Path Sum](#)
- 149. [Count Square Submatrices With All Ones](#)
- 150. [Where Will The Ball Fall](#)

Save Your Time

Save time and master the coding interviews by learning from video lessons at the following link.

<https://algolab.so/p/algorithms-and-data-structure-video-course>

	LeetCode	AlgoExpert	AlgoLab
100+ Practice Questions	✓	✓	✓
Coding Workspace	✓	✓	✓
Multiple Programming Languages	✓	✓	✓
High-Quality Solutions	✓	✓	✓
Video Explanations	✗	✓	✓
All-In-One Platform	✗	✓	✓
Algorithm Patterns Approach	✗	✗	✓
Professional Quality Videos	✗	✗	✓
Value For Money	✗	✗	✓
Private Slack Community	✗	✗	✓
Life-time Access	✗	✗	✓