**DSA Algorithms**

### Searching Algorithms:

1. Linear Search

2. Binary Search

3. Depth-First Search (DFS)

4. Breadth-First Search (BFS)

### Sorting Algorithms:

5. Bubble Sort

6. Selection Sort

7. Insertion Sort

8. Merge Sort

9. Quick Sort

10. Heap Sort

11. Radix Sort

12. Bucket Sort

### Graph Algorithms:

13. Dijkstra's Algorithm

14. Bellman-Ford Algorithm

15. Floyd-Warshall Algorithm

16. Kruskal's Algorithm (Minimum Spanning Tree)

17. Prim's Algorithm (Minimum Spanning Tree)

18. Topological Sorting

19. Tarjan's Algorithm (Strongly Connected Components)

20. Edmonds-Karp Algorithm (Maximum Flow)

21. Ford-Fulkerson Algorithm (Maximum Flow)

### Tree Algorithms:

22. Tree Traversals (Pre-order, In-order, Post-order)

23. Lowest Common Ancestor (LCA)

24. Diameter of a Tree

25. Tree Balancing (AVL, Red-Black Trees)

26. Trie Data Structure

### Dynamic Programming:

27. Fibonacci Sequence (Memoization)

28. Longest Common Subsequence (LCS)

29. Knapsack Problem

30. Coin Change Problem

31. Longest Increasing Subsequence (LIS)

32. Matrix Chain Multiplication

33. Edit Distance

### Greedy Algorithms:

34. Activity Selection

35. Huffman Coding

36. Minimum Spanning Tree (Prim's, Kruskal's)

37. Dijkstra's Algorithm

38. Fractional Knapsack

### Divide and Conquer:

39. Binary Search

40. Merge Sort

41. Quick Sort

42. Strassen's Matrix Multiplication

### Backtracking:

43. N-Queens Problem

44. Sudoku Solver

45. Hamiltonian Cycle

46. Subset Sum

47. Rat in a Maze

### String Algorithms:

48. String Matching (Naive, Rabin-Karp, KMP)

49. Longest Common Substring

50. Manacher's Algorithm (Longest Palindromic Substring)

### Geometric Algorithms:

51. Convex Hull (Graham Scan, Jarvis March)

52. Closest Pair of Points

53. Line Intersection

### Network Flow Algorithms:

54. Ford-Fulkerson Algorithm

55. Edmonds-Karp Algorithm

56. Max Flow Min Cut Theorem

### Randomized Algorithms:

57. Randomized Quick Sort

58. Las Vegas Algorithms

59. Monte Carlo Algorithms

### Combinatorial Algorithms:

60. Permutations

61. Combinations

62. Generating Subsets

### Number Theory Algorithms:

63. Sieve of Eratosthenes

64. Euclidean Algorithm (GCD)

65. Modular Exponentiation

66. Chinese Remainder Theorem

### Data Compression Algorithms:

67. Run-Length Encoding (RLE)

68. Huffman Coding

69. Lempel-Ziv-Welch (LZW) Compression

### Miscellaneous:

70. Subset Sum

71. Josephus Problem

72. Dutch National Flag Problem

73. Longest Palindromic Subsequence

74. All-Pairs Shortest Paths

### Advanced Topics:

75. Persistent Data Structures

76. Suffix Trees and Arrays

77. Segment Trees

78. Fenwick Trees (Binary Indexed Trees)

79. Heavy-Light Decomposition

### Quantum Algorithms:

80. Grover's Algorithm

81. Shor's Algorithm

### Approximation Algorithms:

82. Greedy Set Cover

83. Greedy Vertex Cover

### Online Algorithms:

84. Competitive Analysis

85. Paging Algorithms

### Data Mining and Machine Learning Algorithms:

86. K-means Clustering

87. Apriori Algorithm

### Blockchain Algorithms:

88. Proof of Work (PoW)

89. Proof of Stake (PoS)

### Cryptography Algorithms:

90. RSA Algorithm

91. Elliptic Curve Cryptography (ECC)

### Game Theory Algorithms:

92. Minimax Algorithm

93. Alpha-Beta Pruning

### Distributed Algorithms:

94. Paxos Algorithm

95. Raft Algorithm

### Quantum Computing Algorithms:

96. Quantum Fourier Transform

97. Deutsch-Jozsa Algorithm

### Bioinformatics Algorithms:

98. Sequence Alignment (Needleman-Wunsch, Smith-Waterman)

99. Hidden Markov Models

### Parallel Algorithms:

100. Parallel Prefix (Scan) Algorithm

101. Parallel Merge Sort

### Machine Learning Algorithms:

102. Linear Regression

103. Logistic Regression

104. Decision Trees

105. Support Vector Machines (SVM)

106. Neural Networks (Perceptron, Multi-layer Perceptron)

### Reinforcement Learning Algorithms:

107. Q-Learning

108. Deep Q-Networks (DQN)

### Natural Language Processing (NLP) Algorithms:

109. N-gram Models

110. Hidden Markov Models (HMM)

### Computer Vision Algorithms:

111. Feature Detection (Harris Corner Detection, SIFT, SURF)

112. Object Detection (Haar Cascade, YOLO)

### Recommendation Algorithms:

113. Collaborative Filtering

114. Content-Based Filtering

115. Matrix Factorization

### Sentiment Analysis Algorithms:

116. Naive Bayes Classifier

117. Support Vector Machines (SVM)

118. Recurrent Neural Networks (RNN)

### Graph Neural Networks (GNNs):

119. Graph Convolutional Networks (GCNs)

120. Graph Attention Networks (GATs)

### Reinforcement Learning with Neural Networks:

121. Policy Gradient Methods (REINFORCE, PPO)

122. Actor-Critic Methods (A2C, A3C)

### Multi-agent Reinforcement Learning:

123. Q-Learning with Function Approximation (MARL)

124. Deep Multi-agent Reinforcement Learning (MADDPG)

### Quantum Machine Learning:

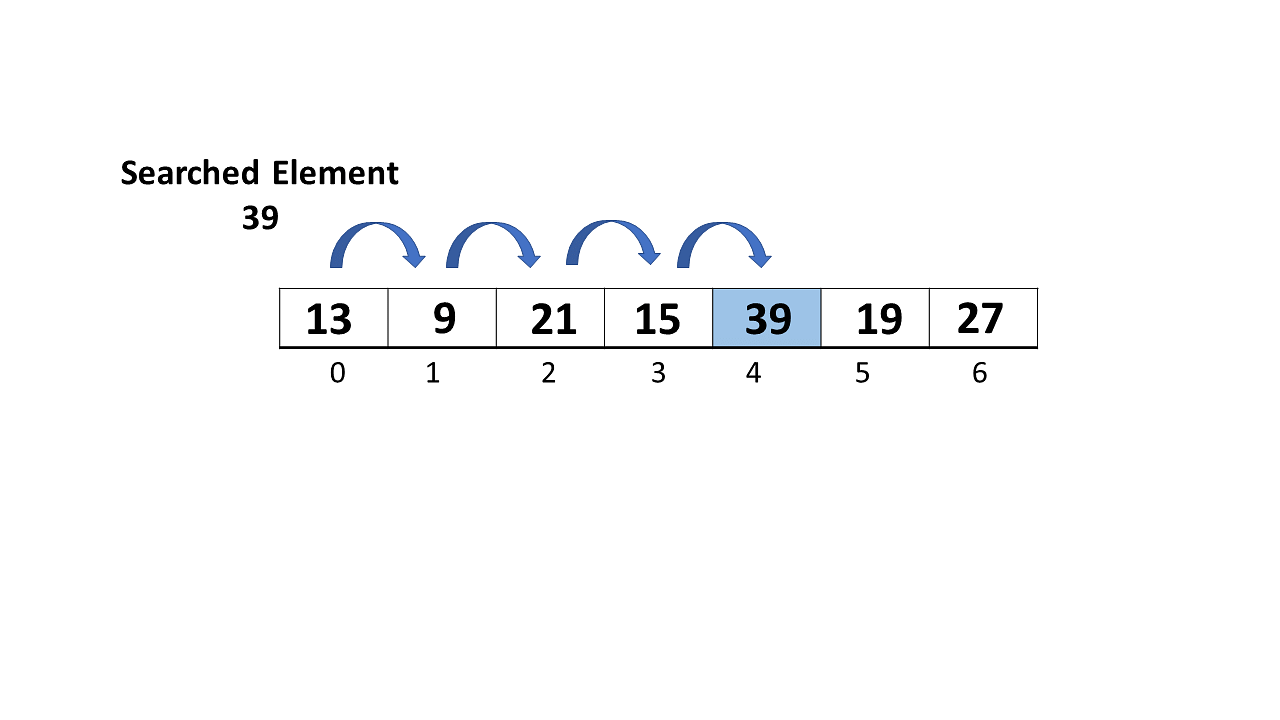
125. Quantum Circuit Learning

126. Quantum Generative Adversarial Networks (QGANs)

127. Quantum Variational Autoencoders (QVAEs)

**1.Linear Search**

**Linear Search** is defined as a sequential [search algorithm](https://www.geeksforgeeks.org/searching-algorithms/) that starts at one end and goes through each element of a list until the desired element is found, otherwise the search continues till the end of the data set.



**How Does Linear Search Algorithm Work?**

In Linear Search Algorithm,

* Every element is considered as a potential match for the key and checked for the same.
* If any element is found equal to the key, the search is successful and the index of that element is returned.
* If no element is found equal to the key, the search yields “No match found”.

**2. Binary Search**

***Binary Search****is defined as a*[*searching algorithm*](https://www.geeksforgeeks.org/searching-algorithms/)*used in a sorted array by****repeatedly dividing the search interval in half****. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(log N).*

**Conditions for when to apply Binary Search in a Data Structure:**

To apply Binary Search algorithm:

* The data structure must be sorted.
* Access to any element of the data structure takes constant time.

