

# DSA SHEET

by

Prashant Kumar

## 90 Days Roadmap

This 90 days Roadmap covers most of the important topics of DSA which are highly recommended for technical interviews and also important to become a good software development engineer. This sheet also covers all the questions topic wise that are needed to be solved in order to crack good product-based companies' interviews.



# ARRAYS

10 Days

## DAY-1

- Understand Big O notation (Time and Space complexity)
- Complexity problems to optimized solution (e.g.-  $O(n^2) \rightarrow O(n)...$ )

## DAY-2

- Basic Concepts of Array like Insertion, Deletion and Searching in array.
- Two pointer approach to solve the array problems.
- Rotate array problem with K steps
- Kadane's Algo to find Max Subarray
- Two Sum Problem
- Buy and Sell Stocks

## DAY - 3

- Understand recursion. Binary Search is must to implement with it.
- Factorial of a number
- Fibonacci number with optimization.
- Length of String using recursion.

## DAY- 4 & 5

- Study Searching and Sorting algorithms thoroughly.
- Quick and Merge Sort is must to have
- Element search in sorted array
- Find smallest letter greater than target
- Search in Rotated Sorted Array
- Merge Sorted array in both orders
- Sort Colors

## DAY - 6

- Searching in 2D Matrix/Array
- Search Matrix Zeros
- Pascals Triangle
- Spiral Matrix 1 and 2
- Set matrix Zeroes
- Rotate Image problem
- Unique Paths 1 and 2

# ARRAYS

10 Days

## DAY-7

- Next Permutations
- Word Search
- Longest Consecutive Sequence in unsorted array
- 3 Sum Problem
- 4 Sum Problem

## DAY- 9

- XOR queries of a subarray.
- Maximum Points you can obtain from cards
- First Missing Positive
- Trapping Rain Water Problem
- Product of array except self
- Container with most water
- Insert Delete get Random- O(1)

## DAY- 8

- Game of Life
- Max Area of Island
- Find all possible Subsets
- Best Time to Buy and Sell Stock 2
- Best Time to Buy and Sell Stock 3
- Maximum Profit from Trading Stocks problem

## DAY – 10

- Pow(x, n)
- Majority Element 1
- Majority Element 2
- Reverse Pairs
- Island Perimeter
- Flood Fill
- Coloring a border

# LINKED LIST

6 Days

## DAY-11

- What is LinkedList? How its different form array and ArrayList?
- When will you choose LinkedList?
- Try to design your own LinkedList with all its major properties like
  - a. Insertion
  - b. Deletion
  - c. LinkedList Traversal
  - d. Searching a Node
- Time and Space complexity for each operation

## DAY- 13

- Swap Nodes in Pairs
- Rotate a given LinkedList
- Remove duplicate elements from sorted LinkedList
- Swap node in the LinkedList without swapping the data
- Move Last element to first position in given LinkedList

## DAY- 12

- Find the length of the LinkedList
- Nth Node of The LinkedList
- Middle of a given LinkedList
- Detect the cycle in LinkedList
- Reverse a LinkedList
- Palindrome of LinkedList
- Intersection of Two LinkedList

## DAY – 14

- Merge K-sorted LinkedList
- Reverse LinkedList Nodes In given K - group
- Subtraction in LinkedList
- Clone a LinkedList with Next and Random Pointer

# LINKED LIST

6 Days

## DAY-15

- Introduction of Doubly LinkedList
- Advantages/Disadvantages of Doubly LinkedList over Singly LinkedList
- Try to design your own Doubly LinkedList with all its major properties like
  - a. Insertion
  - b. Deletion
  - c. Traversal
  - d. Searching a Node
- LRU Cache
- Flatten a multilevel Doubly LinkedList
- Design a Browser History

## DAY- 16

- Introduction to Circular LinkedList
- Implementation of Circular LinkedList with its major features.
- Check if a LinkedList is a Circular LinkedList
- Convert Singly LinkedList in to Circular LinkedList
- Count Nodes in Circular LinkedList
- Check if two Circular LinkedList are identical



# RECURSION

4 Days

## DAY-17

- What is recursion? And how does it make coding simplified?
- Understand the features around which recursion revolves
  - a. Base Case
  - b. How to define Base case
  - c. Recursive calls
- What is Recursion Tree and how does it work internally on memory level.
- Fundamentals of Tail Recursion with tail call elimination for optimization of solutions.

## DAY- 19

- Tower of Hanoi
- Combination Sum 1
- Combination Sum 2
- Recursive function to delete the Kth node of the LinkedList
- Find middle of the LinkedList recursively
- Reverse a doubly LinkedList

## DAY-18

- Fibonacci Number
- Check if a given number is palindrome or not?
- Pascal Triangle
- Permutations

## DAY – 20

- Print all combination of factors of a given number
- Flood Fill
- Word Search
- N-Queens Problem
- Reverse Nodes in K-Group
- Permutation Sequence

# STACK

3 Days

## DAY-21

- Introduction to Stack in Data-Structures with its representations.
- Understand the basic operations of Stack
  - a. PUSH Operation
  - b. POP Operation
  - c. PEEK Operation
  - d. isFull()
  - e. isEmpty()
- Implementation of Stack using Arrays
- Implementation of Stack using Singly LinkedList

## DAY- 22

- Min Stack
- Valid Parentheses
- Convert Infix to Postfix expression
- Convert Prefix to Postfix expression
- The Stock Span Problem
- Daily Temperature
- Next Greater Element

## DAY – 23

- Restrictive Candy Crush
- LRU Cache (Most Important)
- Tweets Count per frequency
- Largest Rectangular area in Histogram
- Implementation of LFU cache
- Find the maximum of minimum for every window size in an array

# QUEUE

5 Days

## DAY-24

- Understand the QUEUE Data Structure
- Understand the basic operations of QUEUE
  - a. ENQUEUE Operation
  - b. DEQUEUE Operation
  - c. PEEK Operation
  - d. isFull()
  - e. isEmpty()
- Implementation of QUEUE using Arrays
- Implementation of Stack using Singly LinkedList

## DAY- 25

- Implementation of QUEUE using STACK
- First Negative Integer in every window of size K
- Valid Substring
- Maximum Diamonds
- Implementation of QUEUE using LinkedList

## DAY – 26

- Reversing first K element of a QUEUE
- Sorting QUEUE without extra Space
- Sliding Window maximum
- Stamping the Sequence
- Minimum Time required to rot all oranges



# QUEUE

5 Days

## DAY-27

- Understand Double Ended Queue (Deque)
- Understand the various operations of Deque Data Structures
  - a. Insertfront()
  - b. InsertLast()
  - c. Deletefront()
  - d. DeleteLast()
- Understand the Circular Queue
- Try out various circular queue operations like Front, Rear.
- Design Circular Deque

## DAY- 28

- Design Circular Queue
- Design Front Middle Back Queue
- Implementation of Deque using doubly LinkedList
- Find the Celebrity

# PRIORITY QUEUE(HEAP)

2 Days

## DAY-29

- Understand the concept of Priority Queue DataStructure
- Understand the difference between Normal Queue and Priority queue
- Understand the various operations of Priority Queue Data Structures
  - a. INSERTION
  - b. DELETION
  - c. PEEK
  - d. POLL
- Implementation of priority Queue using Array

## DAY- 30

- K-Closest points to origin
- Kth Largest Element in an Array
- Maximum Product of two elements in an Array
- Relative Ranks
- Top K frequent Elements
- Adding Array Elements
- Minimum Cost of Ropes
- Binary Heap Operations

# HASHING

4 Days

## DAY-31

- Understand the Hashing and why it is used exactly?
- What is HashCode and how does it get generated?
- Understand the types of Hashing in Data Structure.
- What is Hash Function and user defined Hash Functions
- Understand Index mapping (Trivial Hashing)

## DAY- 33

- Key Pairs
- Top K frequent elements in an Array
- Intersection of Two Arrays
- Design HashMap with its Operations
- Design HashSet with its Operations
- Find an array is subset of another array

## DAY- 32

- What is HashTable and its applications.
- Implement database indexes.
- Used to implement associative arrays.
- Role of HashTable in designing HashMap and HashSet
- Double Hashing and its uses
- Internal Mappings of HashMap/HashSet
- Methods of HashMap/HashSet

## DAY – 34

- Repeated DNA sequence
- Encode & Decode Tiny URL
- Check if two Arrays are equal or not
- Count pairs with given Sum
- Find all triplet with Sum zero
- Longest subarray with sum divisible by K

# BINARY SEARCH

3 Days

## DAY-35

- Implementation of Binary Search
- Search Insert Position
- Intersection of Two Arrays
- Peak Index In a Mountain Array
- Pow (X, N)
- Minimum Limits of Balls in a Bag

## DAY- 36

- Intersection of Two Arrays 2
- Find Smallest Letter Greater Than Target
- Search In a Sorted Array
- Search a 2-D Matrix
- Find the Peak Element
- Minimum Size Subarray Sum
- Kth Smallest Element in a Sorted Matrix

## DAY- 37

- Median of Two Sorted Arrays
- Count of Smaller Number After Self
- Split Array Largest Sum
- Swim In Rising Water
- Shortest Array With Sum At least K
- Count Of Range Sum
- Max Sum of Rectangle No Larger Than K

# TREES

10 Days

## DAY-38

- Understand the Binary Tree DataStructure with its types
- Design Binary tree with Nodes and perform its operations
  - a. Insertion
  - b. Deletion
  - c. Searching
- In-Order Tree Traversal
- Pre-Order Tree Traversal
- Post-Order Tree Traversal
- Level-Order Traversal (with QUEUE)

## DAY- 40

- Invert Binary Tree
- Binary Tree Paths
- Binary Tree Path Sum
- Sub Tree of Another Tree
- Average of levels in Binary Tree
- Evaluate Boolean Binary Tree
- Sum of Left Leaves of given Binary Tree

## DAY- 39

- Maximum Depth of Binary Tree
- Symmetric Tree
- Zig-Zag Level Order Tree Traversal
- Balance Binary Tree
- Diameter of Binary Tree

## DAY – 41

- Flatten Binary Tree to Linked List
- Binary Tree Right Side View
- Left View of a Binary Tree
- Top View of Binary Tree
- Height of a Binary Tree
- Bottom View of Binary Tree
- Sum Tree



# TREES

10 Days

## DAY-42

- Lowest Common Ancestor of a Binary Tree
- Path Sum 2
- Find Largest Value in Each Tree Row
- Binary Tree Maximum Path Sum
- Vertical Order Traversal of a Binary Tree

## DAY- 43

- Understand the Binary Search Tree DataStructure
- Convert Sorted Array to BST
- Search a Node in BST
- Insert a Node in BST
- Delete a Node from BST
- Construct BST from Post-Order Traversal
- Minimum Absolute Difference in BST

## DAY – 44

- Validate Binary Search Tree
- Kth Largest Element in BST
- Predecessor And Successor
- Unique Binary Search Trees
- Unique Binary Search Trees 2
- Convert Sorted List to Binary Search Tree

# TREES

10 Days

## DAY- 45

- Merge Two BSTs
- Kth Smallest Element in BST
- Maximum Sum BST in Binary Tree
- Ceil in BST
- Find Common Nodes in Two BSTs
- Sum of Leaf Nodes in Binary Search Tree

## DAY- 46

- Binary Search Tree Iterator
- Serialize and Deserialize a Binary Search Tree
- Balance a Binary Search Tree
- Remove BST keys outside of the given range
- Binary Tree Cameras
- Number of ways to reorder to get a Binary Search Tree

## DAY - 47

- AVL Tree Insertion
- Optimal Binary Search Tree
- Merge BSTs to create single Binary Search Tree
- Find Shortest Range in Binary Search Tree

# MATH

4 Days

## DAY- 48

- SQRT(X)
- Palindrome Number
- Happy Number
- Power of Two
- Maximum Product of Given Three Numbers

## DAY- 50

- Rearrange an Array with an  $O(1)$  extra Space
- Integer Break
- Power of 2 and Subsequences
- Shuffle an Array
- Brain Game
- Is Binary Number Multiple of 3
- Dice Throw

## DAY- 49

- Minimum Moves to Equal an Array Elements
- Arranging Coins
- Multiply Strings
- Valid Squares
- Basic Calculator 2
- Count Numbers with Unique Digits

## DAY - 51

- Max Points on a Line
- Permutation Sequence
- Number of Digit One
- Count the subarrays having product less than K
- Return Two Prime Numbers
- Poor Pigs

# BIT MANIPULATION

3 Days

## DAY-52

- Count Total Set Bits
- Maximum AND Value
- Missing Number in an Array
- Reverse Bits
- Number of 1 Bits

## DAY- 53

- Find the Duplicate Number
- Josephus Problem
- Maximum XOR of two numbers in an Array
- Division without using multiplication, division and MOD operator

## DAY – 54

- Sum of Bit Difference
- Cinema seat Allocation
- Gray Code 2
- Construct a list using given q XOR queries
- Bleak Numbers

# GRAPH

10 Days

## DAY-55

- Understand GRAPH DataStructure
- Understand Graph Representation
  - a. Adjacency Matrix
  - b. Adjacency List
- Understand both type of Graph
  - a. Directed Graph
  - b. Undirected Graph
- Understand Connected component Graph
- Difference between Tree and Graph and its uses

## DAY-

- Course Schedule (BFS)
- Course Schedule (DFS)
- Topological Sort (BFS)
- Topological Sort (DFS)
- Find the total number of Islands present

## DAY- 56

- Understand and Implement BFS of Graph
- Understand and Implement DFS of Graph
- Possible Paths between Two Vertices
- Sum of Dependencies in a Graph

## DAY – 58

- Understand and Implement Dijkstra Algorithm
- Cheap Flights within K stops
- Understand and implement the Floyd Warshall Algorithm
- Understand and Implement Bellman Ford Algorithm



# GRAPH

10 Days

## DAY-59

- Understand the concept of Spanning Tree and Minimum Spanning Tree
- Read the two most important algorithms of Minimum Spanning tree
  - a. Prims Algorithm
  - b. Kruskal Algorithm
- Implement Minimum Spanning Tree
- Min Cost to Connect all the Points

## DAY- 61

- Numbers of Closed Islands
- Find the City with Smallest Number of Neighbors at a Threshold Distance
- Possible Bipartition
- Find Eventual Safe States
- Is Graph Bipartite?
- Minimum Jumps to Reach Home
- Bridge Edge in a Graph

## DAY- 60

- Johnsons Algorithm (All Pairs Shortest Path)
- Mother Vertex
- Count the paths
- Eulerian Path in an Undirected Graph
- Strongly Connected Components (Kosaraju's Algorithm)

## DAY – 62

- Flood Fill Algorithm
- Clone Graph
- Course Schedule
- Course Schedule 2
- Word Ladder
- Word Ladder 2
- Time needed to inform all employee

# GRAPH

10 Days

## DAY-63

- Minimum number of vertices to reach all Nodes
- Number of Provinces
- Snake and Ladder Problem
- Critical Connections
- Reachable Nodes in Subdivided Graph

## DAY- 64

- Sum of Distance in a Tree
- Cracking the Safe
- Parallel Course 3
- Cheapest Flight with K Stops
- Number of ways to arrive at given Destination

# GREEDY

3 Days

## DAY-65

- Understand the concepts of Greedy Algorithms and its significance
- Implement the Greedy Algorithms like
  - a. Knapsack Problem
  - b. Huffman Coding
  - c. Ford-Fulkerson Algorithm

## DAY- 66

- Number of Coins
- Job Sequencing Problem
- Partition Labels
- Balance a Binary Search Tree
- Reduce Array Size to the Half
- Maximum Total Importance of Road

## DAY- 67

- Remove K- Digits
- Non-Overlapping Intervals
- Candy
- Minimum Number of taps to open to water a garden
- Minimize the heights 2
- Jump Game
- Water Connection Problem

# NUMBER THEORY

3 Days

## DAY-68

- Add Digits
- X of a Kind in a deck of cards
- Find Greatest Common Divisor of an Array
- Remainder of Array Multiplication
- Sum of K Primes

## DAY- 69

- Ugly Number 3
- Simplified Fractions
- Number of pairs of interchangeable rectangles
- Minimum Lines to represent a Line Chart
- Largest sum of Digits in all Divisor
- Find Unit Digit in a Product

## DAY – 70

- Check if it is a good Array
- Count ways to make Array with Product
- Replace non-Coprime Numbers in an Array
- Minimum deletion to make Array Divisible

# DYNAMIC PROGRAMMING

10 Days

## DAY-71

- Understand the fundamentals of Dynamic Programming
- Recursion vs Dynamic Programming
- Understand the Overlapping Subproblem in DP solution
- Tabulation vs Memorization
- Fibonacci Number
- Pascal's Triangle

## DAY- 73

- Get Maximum in Generated Array
- Is Subsequence?
- Stickler Thief
- Longest Repeating Subsequence
- Activity Selection

## DAY- 72

- Counting Bits
- Climbing Stairs
- Nth Fibonacci Number
- Gold Mine Problem
- Best Time to Buy and Sell Stock
- Pascals Triangle 2

## DAY – 74

- Count Number of Hops
- Minimum Falling Path Sum
- Where will the Ball Fall
- Arithmetic Slices
- Beautiful Arrangement
- Best Time to Buy and Sell Stocks 2
- Unique Paths



# DYNAMIC PROGRAMMING

10 Days

## DAY-75

- Box Stacking
- Wildcard Pattern Matching
- Player with Max Score
- Count All Possible paths from Top Left to Bottom Right
- Maximize Dot Product

## DAY- 76

- Count of Strings that can be formed using a, b and c under given constraint
- Maximal Square
- Delete and Earn
- Knight Dialer
- Palindromic Substring
- Word Break

## DAY- 77

- Maximum Product Subarray
- Longest Common Subsequence
- Coin Change 2
- Best Sightseeing Pair
- Best Time to Buy and Sell Stock with Cooldown
- Shopping Offers
- Integer Break

# DYNAMIC PROGRAMMING

10 Days

## DAY-78

- 01 Matrix
- Ugly Number 2
- Matrix Chain Multiplication
- Partition Equal Subset Sum
- Boolean Parenthesization

## DAY-79

- Tapping Rain Water
- Edit Distance
- Best Time to Buy and Sell Stock 4
- Stone Game 4
- Burst Balloons
- Minimum Cost to cut a Stick
- Minimum Difficulty of a Job Schedule

## DAY – 80

- Super Egg Drop
- Shortest Path Visiting all Nodes
- Longest Increasing Path ins a Matrix
- Distinct Subsequences
- Regular Expression Matching
- Longest Valid Parentheses

# STRINGS

6 Days

## DAY-81

- Multiply Two Strings
- Reverse Words in given String
- Group Anagrams
- Integer To Roman
- Implement Trie (Prefix Tree)

## DAY- 83

- Longest K Unique Characters Substring
- Number of Matching Subsequence
- Letter Combination of a Phone Number
- Top K Frequent Words
- Word Subsets
- Decode String
- Find All Anagram in a String

## DAY- 82

- Minimum Remove to make valid Parentheses
- Simplify Path
- Longest Common Prefix in an Array
- Longest Prefix Suffix
- Implement Atoi
- Length of Longest Substring

## DAY – 84

- Permutations of a given String
- Largest Number in K Swaps
- Longest Substring without Repeating Characters
- Reverse Each Words in a Given String
- Add Binary Strings
- Number Following a Pattern

# STRINGS

6 Days

## DAY-85

- Number with One Absolute Difference
- Distinct Palindromic Substrings
- Similar String Group
- Minimum Window Substring
- Count And Say

## DAY – 86

- Find all Possible Palindromic Partition of a String
- Number of distinct word with K maximum contiguous Vowels
- Substring with Concatenation of All Words
- Valid Number
- Text Justification

# BACKTRACKING

2 Days

## DAY-87

- Binary Tree Paths
- Permutations 2
- Combination
- Combination Sum 2
- Letter Case Permutation

## DAY- 88

- Split a String into Max Number of Unique Substrings
- N-Queens 2
- Sudoku Solver
- Rat in a Maze Problem 1
- Largest Number in K Swaps
- Generate Parentheses



# MISC ALGORITHMS

2 Days

## DAY-89

- Counting Sort
- Heap Sort
- Search Pattern (Rabin-Karp Algorithm)
- Pair with given Sum in Sorted Array

## DAY- 90

- Search Pattern (Z-Algorithm)
- Search Pattern (KMP Algorithm)
- Floyd Cycle Detection Algorithm
- Euclid's Algorithm
- Union Find Algorithm
- Moore's Algorithm
- Insert Delete get Random-  $O(1)$



# All the Best



<https://www.linkedin.com/in/prashant-kumar-76b786168/>

<https://github.com/prashantt17>

@prashantkumar