perplexity

Complete 1-Month DSA Learning Roadmap with Python and Tamil Resources

Based on the provided roadmap image and extensive research, here's a comprehensive 30-day plan to master Data Structures and Algorithms (DSA) in Python with free resources and Tamil video tutorials.

[image:1]

Overview

This roadmap follows the structured approach shown in your image, progressing from basic programming concepts to advanced algorithmic techniques. Each day includes **theory learning, coding practice, and problem-solving sessions** with links to free resources and Tamil video tutorials.

Week 1: Foundation Building (Days 1-7)

Day 1: Learn a Programming Language (Python Fundamentals)

Morning Session (2 hours): Python Basics

- **Free Resources:**
  - Python.org Official Tutorial
  - W3Schools Python Tutorial[1]
  - GeeksforGeeks Python Basics[2]
- **Tamil Videos:**
  - Python Data Structures in Tamil - 6 Hour Course[3]
  - Python Programming Tutorial in Tamil[4]
  - Python Tutorial for Beginners in Tamil[5]

Afternoon Session (2 hours): Practice

```
# Basic Python setup for DSA
print('Hello, DSA!')

# Input/Output operations
n = int(input('Enter a number: '))
print(f'You entered: {n}')

# List operations
```

```
numbers = [1, 2, 3, 4, 5]
print(sum(numbers))
```

Evening Session (1 hour): Problem Solving

- **Practice Problems:**
  - Basic input/output operations
  - Variables and data types exercises
  - Control structures (if/else, loops)
  - Functions and recursion basics

Day 2: Arrays & Strings

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Python Arrays[6]
  - Python Lists Documentation
  - W3Schools Python Lists[1]
- **Tamil Videos:**
  - Arrays in Python Tamil Tutorial[7]
  - Data Structures in Tamil - Arrays[8]

Afternoon Session: Implementation

```
# Array operations
arr = [1, 2, 3, 4, 5]
print(arr[0])    # First element
print(arr[-1])   # Last element

# String operations
s = 'hello'
print(s[::-1])   # Reverse string
print(s.upper()) # Uppercase
```

Evening Session: Practice

- **LeetCode Problems:**
  - Two Sum (Easy)
  - Maximum subarray sum
  - Rotate array
- **HackerRank:**
  - Array-DS Problem[9]

Day 3: Linked Lists

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Linked List[2]
  - VisuAlgo Linked List Visualization[10]
  - Algorithm Visualizer[11]
- **Tamil Videos:**
  - Linked List in Tamil - Data Structures[12]
  - Python Linked List Implementation Tamil[3]

Afternoon Session: Implementation

```
# Linked List Node
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

# Basic linked list operations
def create_linked_list(arr):
    if not arr:
        return None
    head = ListNode(arr[0])
    current = head
    for i in range(1, len(arr)):
        current.next = ListNode(arr[i])
        current = current.next
    return head
```

Evening Session: Practice

- **LeetCode Problems:**
  - Reverse a linked list
  - Detect cycle in linked list
  - Merge two sorted linked lists
- **HackerRank:**
  - Linked List challenges[9]

Day 4: Stack & Queue

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Stack and Queue[2]

    - Python collections.deque documentation
    - CSVis Tool Stack Visualization[13]
- **Tamil Videos:**
    - Stack and Queue in Tamil[12]
    - Python Stack Implementation Tamil[3]

Afternoon Session: Implementation

```
# Stack implementation
stack = []
stack.append(1)  # Push
item = stack.pop()  # Pop

# Queue implementation
from collections import deque
queue = deque()
queue.append(1)     # Enqueue
item = queue.popleft()  # Dequeue
```

Evening Session: Practice

- **LeetCode Problems:**
    - Valid parentheses
    - Implement stack using arrays
    - Queue using two stacks
- **Visualization:**
    - AlgoAction Stack/Queue[14]

Day 5: Hashing & Heap

Morning Session: Theory

- **Free Resources:**
    - GeeksforGeeks Hashing[2]
    - Python heapq documentation
    - W3Schools Python Dictionaries[1]
- **Tamil Videos:**
    - Hashing in Tamil - Data Structures[12]
    - Heap implementation Tamil tutorial[3]

Afternoon Session: Implementation

```
# Hash table (dictionary)
hash_map = {}
hash_map['key'] = 'value'
```

```
print(hash_map.get('key'))

# Heap operations
import heapq
heap = []
heapq.heappush(heap, 5)
min_val = heapq.heappop(heap)
```

Evening Session: Practice

- **LeetCode Problems:**
  - Two Sum using hash map
  - Top K frequent elements
  - Find duplicates in array
- **HackerRank:**
  - Heap challenges[9]

Day 6: Recursion & Backtracking

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Recursion[2]
  - Programiz Recursion Tutorial[15]
  - VisuAlgo Recursion Tree[10]
- **Tamil Videos:**
  - Recursion in Tamil - Programming[12]
  - Backtracking Tamil tutorial[3]

Afternoon Session: Implementation

```
# Basic recursion
def factorial(n):
    if n <= 1:
        return 1
    return n * factorial(n - 1)

# Backtracking example
def generate_permutations(arr):
    result = []
    def backtrack(path, remaining):
        if not remaining:
            result.append(path[:])
            return
        for i in range(len(remaining)):
            path.append(remaining[i])
            backtrack(path, remaining[:i] + remaining[i+1:])
            path.pop()
```

```
    backtrack([], arr)
    return result
```

Evening Session: Practice

- **LeetCode Problems:**
  - N-Queens problem
  - Generate parentheses
  - Sudoku solver
- **CodeChef:**
  - Recursion problems[16]

Day 7: Review & Practice

Morning Session: Review

- Revisit all topics from Week 1
- Complete GeeksforGeeks Python DSA Quiz[6]

Afternoon Session: Mixed Practice

- **Contest Problems:**
  - CodeChef Beginner Problems[16]
  - HackerRank Problem Solving[9]

Evening Session: Tamil Video Review

- Complete DSA Tamil Course Review[12]

Week 2: Core Data Structures & Algorithms (Days 8-14)

Day 8: Searching & Sorting

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Searching Algorithms[2]
  - GeeksforGeeks Sorting Algorithms[2]
  - VisuAlgo Sorting Visualization[10]
- **Tamil Videos:**
  - Searching and Sorting in Tamil[12]
  - Binary Search Tamil tutorial[3]

Afternoon Session: Implementation

```
# Binary Search
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1

# Quick Sort
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quick_sort(left) + middle + quick_sort(right)
```

Evening Session: Practice

- **LeetCode Problems:**
  - Binary Search
  - Search in Rotated Sorted Array
  - Merge Sort implementation
- **Visualization:**
  - Sorting Visualizer[17]

Day 9: Mathematical Algorithms

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Mathematical Algorithms[2]
  - Number Theory for Programming[2]
- **Tamil Videos:**
  - Mathematical Algorithms in Tamil[12]

Afternoon Session: Implementation

```
# GCD and LCM
def gcd(a, b):
    while b:
        a, b = b, a % b
```

```
    return a

def lcm(a, b):
    return (a * b) // gcd(a, b)

# Prime number check
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

Evening Session: Practice

- **LeetCode Problems:**
  - Happy Number
  - Ugly Number
  - Count Primes
- **CodeChef:**
  - Mathematical problems[16]

Day 10: Bitwise Operations & Tricks

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Bitwise Algorithms[2]
  - Bit manipulation tricks[2]
- **Tamil Videos:**
  - Bit Manipulation in Tamil[12]

Afternoon Session: Implementation

```
# Common bit operations
def count_set_bits(n):
    count = 0
    while n:
        count += n & 1
        n >>= 1
    return count

def is_power_of_two(n):
    return n > 0 and (n & (n - 1)) == 0

def find_single_number(nums):
    result = 0
    for num in nums:
```

```
        result ^= num
    return result
```

Evening Session: Practice

- **LeetCode Problems:**
  - Single Number
  - Number of 1 Bits
  - Power of Two
- **HackerRank:**
  - Bit Manipulation challenges[9]

Day 11: Trees & Binary Search Trees

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Binary Tree[2]
  - GeeksforGeeks BST[2]
  - VisuAlgo BST Visualization[10]
- **Tamil Videos:**
  - Binary Tree in Tamil[12]
  - BST operations Tamil tutorial[3]

Afternoon Session: Implementation

```
# Binary Tree Node
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

# Tree traversals
def inorder(root):
    if root:
        inorder(root.left)
        print(root.val, end=' ')
        inorder(root.right)

def preorder(root):
    if root:
        print(root.val, end=' ')
        preorder(root.left)
        preorder(root.right)

def postorder(root):
    if root:
```

```
        postorder(root.left)
        postorder(root.right)
        print(root.val, end=' ')
```

Evening Session: Practice

- **LeetCode Problems:**
  - Binary Tree Inorder Traversal
  - Validate Binary Search Tree
  - Maximum Depth of Binary Tree
- **HackerRank:**
  - Tree challenges[9]

Day 12: Greedy Algorithms

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Greedy Algorithms[2]
  - Greedy Algorithm Examples[2]
- **Tamil Videos:**
  - Greedy Algorithms in Tamil[12]

Afternoon Session: Implementation

```
# Activity Selection Problem
def activity_selection(start, finish):
    n = len(finish)
    activities = list(range(n))
    activities.sort(key=lambda x: finish[x])

    selected = [activities[0]]
    last_selected = 0

    for i in range(1, n):
        if start[activities[i]] >= finish[activities[last_selected]]:
            selected.append(activities[i])
            last_selected = i

    return selected

# Fractional Knapsack
def fractional_knapsack(weights, values, capacity):
    n = len(weights)
    items = [(values[i]/weights[i], weights[i], values[i]) for i in range(n)]
    items.sort(reverse=True)

    total_value = 0
    for ratio, weight, value in items:
```

```
        if capacity >= weight:
            capacity -= weight
            total_value += value
        else:
            total_value += capacity * ratio
            break

    return total_value
```

Evening Session: Practice

- **LeetCode Problems:**
  - Jump Game
  - Gas Station
  - Minimum Number of Arrows
- **CodeChef:**
  - Greedy problems[16]

Day 13: Dynamic Programming Basics

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Dynamic Programming[2]
  - DP Patterns[2]
- **Tamil Videos:**
  - Dynamic Programming in Tamil[12]

Afternoon Session: Implementation

```
# Fibonacci with memoization
def fibonacci(n, memo={}):
    if n in memo:
        return memo[n]
    if n <= 1:
        return n
    memo[n] = fibonacci(n-1, memo) + fibonacci(n-2, memo)
    return memo[n]

# Coin Change Problem
def coin_change(coins, amount):
    dp = [float('inf')] * (amount + 1)
    dp[0] = 0

    for coin in coins:
        for i in range(coin, amount + 1):
            dp[i] = min(dp[i], dp[i - coin] + 1)
```

```
    return dp[amount] if dp[amount] != float('inf') else -1
```

Evening Session: Practice

- **LeetCode Problems:**
  - Climbing Stairs
  - House Robber
  - Coin Change
- **HackerRank:**
  - DP challenges[9]

Day 14: Graph Algorithms Basics

Morning Session: Theory

- **Free Resources:**
  - GeeksforGeeks Graph Data Structure[2]
  - Graph Traversal Algorithms[2]
  - VisuAlgo Graph Visualization[10]
- **Tamil Videos:**
  - Graph Algorithms in Tamil[12]
  - BFS and DFS Tamil tutorial[3]

Afternoon Session: Implementation

```
# Graph representation
from collections import defaultdict, deque

class Graph:
    def __init__(self):
        self.graph = defaultdict(list)

    def add_edge(self, u, v):
        self.graph[u].append(v)

    def bfs(self, start):
        visited = set()
        queue = deque([start])
        visited.add(start)

        while queue:
            vertex = queue.popleft()
            print(vertex, end=' ')

            for neighbor in self.graph[vertex]:
                if neighbor not in visited:
                    visited.add(neighbor)
```

```
                    queue.append(neighbor)

    def dfs(self, start, visited=None):
        if visited is None:
            visited = set()

        visited.add(start)
        print(start, end=' ')

        for neighbor in self.graph[start]:
            if neighbor not in visited:
                self.dfs(neighbor, visited)
```

Evening Session: Practice

- **LeetCode Problems:**
  - Number of Islands
  - Clone Graph
  - Course Schedule
- **HackerRank:**
  - Graph challenges[9]

Week 3: Advanced Algorithms & Practice (Days 15-21)

Day 15-21: Advanced Topics & Mixed Practice

Daily Structure:

- **Morning (2 hours):** Advanced topic study
- **Afternoon (2 hours):** Implementation and coding
- **Evening (1 hour):** Contest problems and review

Advanced Topics Coverage:

- **Day 15:** Advanced Dynamic Programming
- **Day 16:** Graph Algorithms (Dijkstra, MST)
- **Day 17:** Advanced Data Structures (Trie, Segment Tree)
- **Day 18:** String Algorithms
- **Day 19:** Advanced Graph Algorithms
- **Day 20:** Competitive Programming Techniques
- **Day 21:** Mock Interviews and Review

Week 4: Final Practice & Assessment (Days 22-30)

Daily Practice Sessions

Platform-wise Practice:

- **LeetCode:** Interview Crash Course[18]
- **HackerRank:** Data Structures Domain[9]
- **CodeChef:** Python Programming Practice[16]
- **Coderbyte:** Python Challenges[19]

Free Visualization Tools:

- **Algorithm Visualizer:** https://algorithm-visualizer.org[11]
- **VisuAlgo:** https://visualgo.net[10]
- **AlgoAction:** https://algoaction.netlify.app[14]
- **CSVis Tool:** https://csvistool.com[13]

Key Tamil Video Resources

Comprehensive Tamil Courses:

1. Master Python Data Structures in Tamil - 6 Hour Course[3] - Complete Python DSA course
2. Data Structures and Algorithms Full Course in Tamil[12] - 9+ hour comprehensive course
3. Python Data Structures & Algorithms in Tamil[5] - 30-minute quick overview
4. Data Structures in Tamil Playlist[8] - Topic-wise tutorials

Free Practice Platforms

Primary Platforms:

- **LeetCode:** https://leetcode.com[18] - Industry standard problems
- **HackerRank:** https://www.hackerrank.com[9] - Structured challenges
- **CodeChef:** https://www.codechef.com[16] - Competitive programming
- **GeeksforGeeks:** https://practice.geeksforgeeks.org[2] - Practice problems
- **HackerEarth:** https://www.hackerearth.com[20] - Coding challenges
- **Coderbyte:** https://coderbyte.com[19] - Python specific challenges

Key Code Examples Repository

Essential Python DSA Implementations:

```
# Complete DSA toolkit
# Available at: https://github.com/Python30Days (inspired by)[84]
```

```
# 1. Basic Data Structures
class Array:
    def __init__(self):
        self.data = []

    def insert(self, index, value):
        self.data.insert(index, value)

    def search(self, value):
        return value in self.data

# 2. Linked List
class LinkedList:
    def __init__(self):
        self.head = None

    def insert(self, data):
        new_node = ListNode(data)
        new_node.next = self.head
        self.head = new_node

# 3. Stack & Queue implementations
# 4. Tree traversals
# 5. Graph algorithms
# 6. Sorting algorithms
# 7. Dynamic programming solutions
```

Success Tips

1. **Daily Consistency:** Follow the schedule religiously for 30 days
2. **Tamil Videos:** Use Tamil explanations to understand concepts clearly
3. **Visualization:** Leverage the mentioned visualization tools
4. **Practice Problems:** Solve at least 5 problems daily
5. **Code Review:** Review and optimize your solutions
6. **Mock Interviews:** Practice explaining your solutions

Free Resources Summary

Books & Documentation:

- Python Official Documentation
- GeeksforGeeks Complete DSA Guide[2]
- Algorithm Design Manual (Free PDF)[2]

Online Courses:

- Jovian Data Structures and Algorithms[21]
- Programiz DSA Course[15]
- W3Schools DSA with Python[1]

This comprehensive 30-day roadmap provides everything needed to master DSA in Python with free resources and Tamil video support. Follow the daily schedule consistently, practice regularly, and use the visualization tools to understand concepts clearly.