# AI-SDE & Data Structures and Algorithms (DSA) Roadmap

## Phase 1: Basics of Programming (1-2 Weeks)

### Topics to Learn:

* Variables & Data Types
* Conditional Statements (if-else)
* Loops (for, while)
* Functions & Recursion
* Time Complexity (Big-O Notation)

### AI-SDE Add-ons:

* Introduction to Python for AI (NumPy, Pandas, Matplotlib)
* Basics of SQL for Data Handling

### Practice Problems:

* Print prime numbers in a range
* Check if a number is palindrome
* Factorial using recursion
* Fibonacci series

### Resources:

* [Learn Python Basics](https://www.w3schools.com/python/)
* [Learn C++ Basics](https://www.learncpp.com/)
* [Python for Data Science - Kaggle](https://www.kaggle.com/learn/python)

## Phase 2: Arrays & Strings (2-3 Weeks)

### Topics to Learn:

* Arrays (1D, 2D) – Traversal, Insertion, Deletion
* Searching – Linear & Binary Search
* Sorting – Bubble, Selection, Insertion, Merge, Quick Sort
* Strings – Reversal, Palindrome, Anagram
* Two-Pointer Technique

### AI-SDE Add-ons:

* Data Preprocessing Techniques (Handling Missing Data, Encoding)

### Practice Problems:

* Find the second-largest element in an array
* Rotate an array by k positions
* Sort an array using Quick Sort
* Check if two strings are anagrams

### Resources:

* [Array Problems - LeetCode](https://leetcode.com/problemset/all/?topicSlugs=array)
* [Sorting Algorithms - GeeksforGeeks](https://www.geeksforgeeks.org/sorting-algorithms/)
* [Data Preprocessing - Scikit-Learn](https://scikit-learn.org/stable/modules/preprocessing.html)

## Phase 3: Linked Lists (2 Weeks)

### Topics to Learn:

* Singly Linked List – Insertion, Deletion, Reversal
* Doubly Linked List – Insertion, Deletion
* Cycle Detection (Floyd’s Algorithm)
* Merge Two Sorted Linked Lists

### AI-SDE Add-ons:

* Introduction to Machine Learning (Supervised vs Unsupervised Learning)
* Basics of Scikit-Learn

### Practice Problems:

* Reverse a linked list
* Detect a cycle in a linked list
* Find the middle element of a linked list

### Resources:

* [Linked List Problems - LeetCode](https://leetcode.com/tag/linked-list/)
* [Linked List Guide - GFG](https://www.geeksforgeeks.org/data-structures/linked-list/)
* [Machine Learning with Scikit-Learn](https://scikit-learn.org/stable/tutorial/basic/tutorial.html)

## Phase 4: Stacks & Queues (2 Weeks)

### Topics to Learn:

* Stack – Push, Pop, Peek (LIFO)
* Queue – Enqueue, Dequeue (FIFO)
* Implement Stack using Queue (and vice versa)
* Next Greater Element

### AI-SDE Add-ons:

* Basics of Neural Networks (Perceptron, Activation Functions)
* Implementing a Simple Neural Network in Python

### Practice Problems:

* Implement a stack using an array
* Find the next greater element for each element in an array
* Implement a queue using two stacks

### Resources:

* [Stack Problems - LeetCode](https://leetcode.com/tag/stack/)
* [Queue Problems - GFG](https://www.geeksforgeeks.org/queue-data-structure/)
* [Neural Networks - Stanford CS231n](http://cs231n.github.io/neural-networks-1/)

## Phase 5: Recursion & Backtracking (2 Weeks)

### Topics to Learn:

* Recursion – Factorial, Fibonacci, Tower of Hanoi
* Backtracking – N-Queens, Rat in a Maze

### AI-SDE Add-ons:

* Introduction to Deep Learning
* Hands-on with TensorFlow & PyTorch

### Practice Problems:

* Solve N-Queens problem
* Generate all subsets of a set
* Print all permutations of a string

### Resources:

* [Recursion & Backtracking - LeetCode](https://leetcode.com/tag/recursion/)
* [Deep Learning with TensorFlow](https://www.tensorflow.org/tutorials)

## Phase 6: Trees & Graphs (4 Weeks)

### Topics to Learn:

* Trees – Binary Tree, BST, Tree Traversals (Preorder, Inorder, Postorder)
* Graphs – BFS, DFS, Dijkstra’s Algorithm

### AI-SDE Add-ons:

* Implementing AI Algorithms (Decision Trees, Random Forests)

### Practice Problems:

* Find the height of a binary tree
* Check if a tree is balanced
* Find the shortest path in a graph using Dijkstra’s algorithm

### Resources:

* [Tree Problems - LeetCode](https://leetcode.com/tag/tree/)
* [Graph Problems - GFG](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/)

## Phase 7: Advanced Topics & AI Deployment (4 Weeks)

### Topics to Learn:

* Dynamic Programming
* Greedy Algorithms
* Cloud Deployment (AWS, Docker, Kubernetes)

### AI-SDE Add-ons:

* Model Deployment with Flask/FastAPI
* MLOps & Model Optimization

### Practice Problems:

* Solve 0/1 Knapsack problem
* Deploy an ML model as a REST API

### Resources:

* [MLOps Guide - GFG](https://www.geeksforgeeks.org/machine-learning-operations-mlops/)

## Final Phase: Competitive Programming & AI Project Development

* Participate in contests on **CodeForces, AtCoder, CodeChef**
* Solve **previously asked interview questions** from top tech companies
* Build an AI-powered project (**Computer Vision, NLP, Predictive Modeling**)

### Resources:

* [Top Interview Questions - LeetCode](https://leetcode.com/explore/interview/)
* [Deploying AI Models - TensorFlow](https://www.tensorflow.org/tfx)

## How to Stay Consistent?

* Solve **at least 3-5 problems daily**
* Work on **real-world AI projects**
* Track progress using [LeetCode Progress Sheet](https://leetcode.com/)

## Summary:

1. Master **DSA fundamentals**
2. Learn **AI frameworks & model deployment**
3. Work on **real-world AI-SDE projects**
4. Keep solving **competitive programming problems**
5. **Never stop learning!** 🚀