

Data Structures, Algorithms & System Design Course

ისწავლე, მოემზადე ტექნიკური ინტერვიუებისთვის და [დასაქმდი](#) მოწინავე ტექ კომპანიებში, როგორიცაა FAANG კომპანიები - Meta (ყოფილი Facebook), Amazon, Apple, Netflix და Google.



Data Structures, Algorithms & System Design Course

Consists of:

- 1. Data Structures & Algorithms**
- 2. System Design**

Data Structures & Algorithms

Data Structures & Algorithms is a field of study that involves understanding how to organize, manage, and manipulate data efficiently, enabling more complex computational tasks to be performed, and it's essential for programmers to learn as it forms the foundation of efficient problem-solving, enabling the creation of high-performance, scalable software, and is often a key aspect of technical interviews in the software industry.

Part 0

Topic	Priority
Python Basics	High
Time & Space Complexity	High

Part 1

Topic	Priority
Array	High
String	High
Dictionary (Hashing)	Mid
Recursion	Mid

Part 2

Topic	Priority
Sorting and searching	High
Matrix	High
Queue	Mid
Stack	Mid

Part 3

Topic	Priority
Tree (BT, BST)	High
Linked List	Mid
Graph	High
Trie	Mid

Part 4

Topic	Priority
Dynamic programming (DFS, BFS)	Low
Binary + Logical Operators	Low
Interval	Low
Math & Geometry	Low

Part 5

Problem	Difficulty
Two Sum	Easy
Valid Parentheses	Easy

Problem	Difficulty
Merge Two Sorted Lists	Easy
Best Time to Buy and Sell Stock	Easy
Valid Palindrome	Easy
Invert Binary Tree	Easy
Valid Anagram	Easy
Binary Search	Easy
Flood Fill	Easy
Lowest Common Ancestor of a Binary Search Tree	Easy
Balanced Binary Tree	Easy
Linked List Cycle	Easy

Part 6

Problem	Difficulty
Implement Queue using Stacks	Easy
First Bad Version	Easy
Ransom Note	Easy
Climbing Stairs	Easy
Longest Palindrome	Easy
Reverse Linked List	Easy
Majority Element	Easy
Add Binary	Easy
Diameter of Binary Tree	Easy
Middle of the Linked List	Easy
Maximum Depth of Binary Tree	Easy
Contains Duplicate	Easy

Part 7

Problem	Difficulty
Min Stack	Medium
Maximum Subarray	Medium
Insert Interval	Medium
01 Matrix	Medium
K Closest Points to Origin	Medium
Longest Substring Without Repeating Characters	Medium
3Sum	Medium
Binary Tree Level Order Traversal	Medium
Clone Graph	Medium
Evaluate Reverse Polish Notation	Medium

Part 8

Problem	Difficulty
Course Schedule	Medium
Implement Trie (Prefix Tree)	Medium
Coin Change	Medium
Product of Array Except Self	Medium
Validate Binary Search Tree	Medium
Number of Islands	Medium
Rotting Oranges	Medium
Search in Rotated Sorted Array	Medium

Part 9

Problem	Difficulty
Combination Sum	Medium
Permutations	Medium
Merge Intervals	Medium
Lowest Common Ancestor of a Binary Tree	Medium
Time Based Key-Value Store	Medium
Accounts Merge	Medium
Sort Colors	Medium
Word Break	Medium

Part 10

Problem	Difficulty
Partition Equal Subset Sum	Medium
String to Integer (atoi)	Medium
Spiral Matrix	Medium
Subsets	Medium
Binary Tree Right Side View	Medium
Longest Palindromic Substring	Medium
Unique Paths	Medium
Construct Binary Tree from Preorder and Inorder Traversal	Medium
Container With Most Water	Medium

Part 11

Problem	Difficulty
Letter Combinations of a Phone Number	Medium
Word Search	Medium
Find All Anagrams in a String	Medium
Minimum Height Trees	Medium
Task Scheduler	Medium
LRU Cache	Medium
Kth Smallest Element in a BST	Medium
Minimum Window Substring	Hard

Part 12

Problem	Difficulty
Serialize and Deserialize Binary Tree	Hard
Trapping Rain Water	Hard
Find Median from Data Stream	Hard
Word Ladder	Hard
Basic Calculator	Hard
Maximum Profit in Job Scheduling	Hard
Merge k Sorted Lists	Hard
Largest Rectangle in Histogram	Hard

System Design

System Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements, often focusing on high-level design like system architecture and database design, and it's crucial for programmers to learn as it enables them to understand and create efficient, scalable, and maintainable systems, and often forms a significant part of the decision-making process in software development and technical interviews.

Introduction

- What is System Design?
- How To: System Design?

Performance vs Scalability

Latency vs Throughput

Availability vs Consistency

- CAP Theorem
 - CP - consistency and partition tolerance
 - AP - availability and partition tolerance

Consistency Patterns

- Weak consistency
- Eventual Consistency
- Strong consistency

Availability Patterns

- Fail-Over
 - Active-passive

- Active-active
- Availability in Numbers
 - 99.9% Availability - Three 9s
 - 99.99% Availability - Four 9s
 - Availability in parallel vs in sequence

Replication

- Master-Master replication
- Master-Slave replication

Background Jobs

- Event Driven
- Schedule Driven
- Returning Results

Domain Name System (DNS)

Content Delivery Networks (CDN)

- Push CDNs
- Pull CDNs

Load Balancers

- Load Balancer vs Reverse Proxy
- Load Balancing Algorithms
- Layer 7 Load Balancing
- Layer 4 Load Balancing
- Horizontal Scaling

Cloud

- AWS
- Azure
- Google Cloud

Application Layer

- Microservices
- Service Discovery

Databases

- SQL vs noSQL
- NoSQL
 - Key Value Store
 - Document Store
 - Wide Column Store
 - Graph Databases
- RDBMS
 - Replication
 - Sharding
 - Federation
 - Denormalization
 - SQL Tuning

Caching

- Client Caching
- CDN Caching
- Web Server Caching
- Database Caching
- Application Caching

Asynchronism

- Back Pressure
- Task Queues
- Message Queues (SQS, DLQ)

Idempotent Operations

Communication

- Networking protocols: HTTP, TCP, UDP.
- Architectural styles: RPC, REST, and gRPC.

Security

System Design Examples

- Online Library, Url Shortener, Instagram live