



# Complete DSA Project Roadmap: Start to FAANG Ready

Based on comprehensive research and progressive learning principles, here's your structured project sequence from absolute beginner to FAANG interview-ready:

## Phase 1: Foundation Building (Months 1-2)

*Time Investment: 1-2 hours daily*

### Week 1-2: Programming Fundamentals + Basic Data Structures

#### 1. Number Guessing Game with Statistics<sup>[1]</sup>

- **Focus:** Basic arrays, loops, conditionals
- **Duration:** 3-4 days
- **DSA Concepts:** Linear search, basic sorting

#### 2. Simple Calculator with History<sup>[2]</sup>

- **Focus:** Stack implementation (LIFO operations)
- **Duration:** 3-4 days
- **DSA Concepts:** Stack operations, basic arithmetic algorithms

### Week 3-4: Linear Data Structures

#### 3. To-Do List Application<sup>[2]</sup> <sup>[1]</sup>

- **Focus:** Dynamic arrays, linked lists
- **Duration:** 5-7 days
- **DSA Concepts:** CRUD operations, list manipulation

#### 4. Student Grade Management System<sup>[3]</sup> <sup>[2]</sup>

- **Focus:** Arrays, basic sorting algorithms
- **Duration:** 5-7 days
- **DSA Concepts:** Bubble sort, selection sort, linear search

## Week 5-8: Hash Tables and Trees

### 5. Library Management System<sup>[4]</sup> <sup>[5]</sup> <sup>[6]</sup>

- **Focus:** Hash tables, linked lists, file I/O
- **Duration:** 10-14 days
- **DSA Concepts:** Hash functions, collision handling, basic CRUD

### 6. Binary Search Tree Implementation<sup>[7]</sup> <sup>[8]</sup>

- **Focus:** Tree traversals, recursion
- **Duration:** 7-10 days
- **DSA Concepts:** BST operations, DFS traversals

## Phase 2: Intermediate Skills (Months 2-4)

*Time Investment: 2-3 hours daily*

## Week 9-12: Algorithm Visualization

### 7. Sorting Algorithm Visualizer<sup>[6]</sup> <sup>[9]</sup>

- **Focus:** All major sorting algorithms
- **Duration:** 14-21 days
- **DSA Concepts:** Merge sort, quick sort, heap sort, time complexity analysis

### 8. Maze Solver<sup>[6]</sup>

- **Focus:** Graph traversal algorithms
- **Duration:** 10-14 days
- **DSA Concepts:** BFS, DFS, pathfinding

## Week 13-16: Advanced Data Structures

### 9. Spell Checker with Trie<sup>[10]</sup>

- **Focus:** Trie implementation, string algorithms
- **Duration:** 14-21 days
- **DSA Concepts:** Trie construction, string matching, prefix operations

### 10. File Compression Tool (Huffman Coding)<sup>[4]</sup> <sup>[10]</sup>

- **Focus:** Binary heaps, tree structures
- **Duration:** 21-28 days
- **DSA Concepts:** Priority queues, Huffman algorithm, bit manipulation

## Phase 3: Advanced Intermediate (Months 4-6)

*Time Investment: 3-4 hours daily*

### Week 17-20: Graph Algorithms

#### 11. Social Network Friend Recommendation<sup>[5]</sup> <sup>[4]</sup>

- **Focus:** Graph representations, traversal optimization
- **Duration:** 21-28 days
- **DSA Concepts:** Adjacency lists/matrices, BFS/DFS, shortest paths

#### 12. Travel Route Planner<sup>[2]</sup>

- **Focus:** Weighted graphs, shortest path algorithms
- **Duration:** 21-28 days
- **DSA Concepts:** Dijkstra's algorithm, A\* search, graph optimization

### Week 21-24: Dynamic Programming

#### 13. Stock Price Analysis System<sup>[4]</sup>

- **Focus:** Time series analysis, optimization algorithms
- **Duration:** 21-28 days
- **DSA Concepts:** Sliding window, DP patterns, heap operations

#### 14. Text Analysis Tool (LCS, Edit Distance)<sup>[8]</sup>

- **Focus:** String DP algorithms
- **Duration:** 14-21 days
- **DSA Concepts:** Longest Common Subsequence, Edit Distance, DP optimization

## Phase 4: FAANG Interview Preparation (Months 6-9)

*Time Investment: 4-5 hours daily*

### Months 6-7: System Design Fundamentals

#### 15. Distributed Cache System (Mini-Redis)<sup>[11]</sup> <sup>[12]</sup>

- **Focus:** System design, concurrent data structures
- **Duration:** 4-6 weeks
- **DSA Concepts:** LRU/LFU cache, consistent hashing, thread safety

#### 16. Real-Time Chat Application<sup>[5]</sup> <sup>[4]</sup>

- **Focus:** Real-time systems, message ordering
- **Duration:** 4-6 weeks
- **DSA Concepts:** Priority queues, graph algorithms, efficient data structures

## Month 7-8: Advanced System Projects

### 17. Search Engine with Web Crawler<sup>[11]</sup> <sup>[2]</sup>

- **Focus:** Large-scale data processing
- **Duration:** 6-8 weeks
- **DSA Concepts:** Graph algorithms (BFS), inverted index, PageRank algorithm

### 18. Code Version Control System (Mini-Git)<sup>[13]</sup> <sup>[14]</sup>

- **Focus:** Complex graph algorithms, file systems
- **Duration:** 6-8 weeks
- **DSA Concepts:** Merkle trees, graph traversal, diff algorithms

## Month 8-9: Specialization Projects

### 19. Machine Learning Framework<sup>[13]</sup>

- **Focus:** Mathematical algorithms, optimization
- **Duration:** 6-8 weeks
- **DSA Concepts:** Matrix operations, gradient descent, decision trees

### 20. Database Engine Implementation<sup>[11]</sup>

- **Focus:** Storage systems, query optimization
- **Duration:** 8-10 weeks
- **DSA Concepts:** B+ trees, query algorithms, transaction management

## Phase 5: FAANG Interview Mastery (Months 9-12)

*Time Investment: 5-6 hours daily*

## Advanced Specialization (Choose 2-3 based on target company)

### For Google:

#### 21. Blockchain Implementation<sup>[5]</sup>

- **Duration:** 8-10 weeks
- **DSA Concepts:** Hash chains, consensus algorithms, P2P networking

### For Facebook/Meta:

#### 22. Recommendation Engine

- **Duration:** 6-8 weeks
- **DSA Concepts:** Collaborative filtering, graph algorithms, ML algorithms

### For Amazon:

#### 23. E-commerce Order Processing System<sup>[15]</sup>

- **Duration:** 8-10 weeks
- **DSA Concepts:** Distributed systems, inventory algorithms, optimization

## For All Companies:

### 24. Ride-Sharing Service (Uber Clone) <sup>[12]</sup> <sup>[11]</sup>

- **Duration:** 8-12 weeks
- **DSA Concepts:** Geospatial algorithms, matching algorithms, real-time optimization

## Timeline Summary

Phase	Duration	Focus	Daily Hours	Projects Completed
Foundation	Months 1-2	Basic DSA <sup>[16]</sup> <sup>[7]</sup>	1-2 hours	6 projects
Intermediate	Months 2-4	Core Algorithms <sup>[7]</sup> <sup>[17]</sup>	2-3 hours	4 projects
Advanced Intermediate	Months 4-6	Complex DSA <sup>[8]</sup> <sup>[18]</sup>	3-4 hours	4 projects
FAANG Prep	Months 6-9	System Design <sup>[11]</sup> <sup>[12]</sup>	4-5 hours	4-6 projects
Interview Mastery	Months 9-12	Specialization <sup>[13]</sup> <sup>[19]</sup>	5-6 hours	2-4 projects

## Key Learning Milestones

### Month 2 Checkpoint:

- Can implement all basic data structures from scratch <sup>[7]</sup>
- Solved 50+ easy-level problems on LeetCode <sup>[18]</sup>
- Understanding of time/space complexity <sup>[16]</sup> <sup>[7]</sup>

### Month 4 Checkpoint:

- Master sorting and searching algorithms <sup>[17]</sup> <sup>[7]</sup>
- Solved 75+ medium-level problems <sup>[17]</sup>
- Can explain algorithm trade-offs clearly

### Month 6 Checkpoint:

- Proficient in graph algorithms and dynamic programming <sup>[8]</sup> <sup>[18]</sup>
- Solved 100+ medium-level problems
- Can design basic systems architecture

### Month 9 Checkpoint:

- Can handle system design interviews <sup>[12]</sup> <sup>[11]</sup>
- Solved 50+ hard-level problems
- Ready for FAANG technical rounds <sup>[20]</sup>

## Month 12 Checkpoint:

- Expert-level DSA knowledge<sup>[16]</sup>
- Complete portfolio of 20+ projects
- FAANG interview ready<sup>[21] [22]</sup>

## Success Metrics by Phase

### Beginner Success (Months 1-2):

- Complete understanding of arrays, linked lists, stacks, queues<sup>[7]</sup>
- Can implement basic sorting algorithms<sup>[17]</sup>
- Solved 50+ easy problems consistently

### Intermediate Success (Months 2-4):

- Proficient in trees, graphs, and hash tables<sup>[8]</sup>
- Understanding of recursive algorithms<sup>[7]</sup>
- Can solve medium-level problems in 30-45 minutes

### Advanced Success (Months 4-6):

- Master dynamic programming and greedy algorithms<sup>[8] [17]</sup>
- Can design efficient solutions for complex problems
- Consistent problem-solving speed improvement

### FAANG Ready Success (Months 6-12):

- Expert in system design principles<sup>[12] [11]</sup>
- Can solve hard problems in interview timeframes
- Strong project portfolio demonstrating real-world applications<sup>[22] [13]</sup>

This progressive roadmap ensures you build **strong fundamentals** before advancing to **complex system design**, making you well-prepared for FAANG interviews while developing practical skills through **real-world projects**.<sup>[23] [20] [21]</sup>

✱

1. <https://writool.com/news/dsa-project-ideas/>
2. <https://codegnan.com/data-structure-and-algorithm-project-ideas/>
3. <https://www.geeksforgeeks.org/cpp/top-50-cpp-project-ideas-for-beginners-advanced/>
4. <https://www.guvi.in/blog/data-structures-and-algorithms-project-ideas/>
5. <https://www.fynd.academy/blog/dsa-projects-ideas>
6. <https://www.upgrad.com/blog/data-structure-project-ideas-beginners/>

7. <https://www.bosscoderacademy.com/blog/complete-dsa-roadmap>
8. <https://dev.to/iuliagroza/complete-introduction-to-the-30-most-essential-data-structures-algorithms-43kd>
9. <https://www.simplilearn.com/tutorials/data-structure-tutorial/best-dsa-projects>
10. <https://iq.opengenus.org/list-of-algorithmic-project-ideas/>
11. <https://www.geeksforgeeks.org/system-design/system-design-interviews-faang/>
12. <https://www.tryexponent.com/blog/system-design-interview-guide>
13. [https://www.linkedin.com/posts/srishtik-dutta-890587160\\_faangprep-resumetips-projectbuilding-activity-7325104478066507776-9qTK](https://www.linkedin.com/posts/srishtik-dutta-890587160_faangprep-resumetips-projectbuilding-activity-7325104478066507776-9qTK)
14. [https://www.reddit.com/r/cscareerquestions/comments/p6ospv/projects\\_to\\_showcase\\_data\\_structures\\_algorithms/](https://www.reddit.com/r/cscareerquestions/comments/p6ospv/projects_to_showcase_data_structures_algorithms/)
15. <https://igotanoffer.com/blogs/tech/amazon-coding-interview-questions>
16. <https://www.designgurus.io/answers/detail/how-many-months-to-prepare-dsa>
17. <https://tutetude.com/blogs/mastering-data-structures-algorithms-guide/>
18. <https://dev.to/somadevtoo/the-software-developer-interview-prep-roadmap-with-resources-332l>
19. <https://blog.devgenius.io/how-did-i-crack-a-bunch-of-faangs-and-how-you-could-do-it-too-d824cfc6b974>
20. <https://www.geeksforgeeks.org/blogs/how-to-prepare-for-faang-interviews-faang-interview-prep-plan/>
21. <https://www.cromacampus.com/blogs/how-to-get-a-tech-job-at-maang-companies/>
22. <https://pesto.tech/resources/how-to-get-a-tech-job-at-maang-companies>
23. <https://blog.heycoach.in/a-guide-to-get-placed-in-maang-companies/>
24. <https://www.mta.ca/~rrosebru/oldcourse/263114/Dsa.pdf>
25. <https://www.geeksforgeeks.org/complete-roadmap-to-learn-dsa-from-scratch/>
26. <https://www.geeksforgeeks.org/dsa/complete-roadmap-to-learn-dsa-from-scratch/>
27. [https://www.reddit.com/r/learnprogramming/comments/14e52ia/learning\\_dsa\\_from\\_scratch\\_the\\_ultimate\\_guide/](https://www.reddit.com/r/learnprogramming/comments/14e52ia/learning_dsa_from_scratch_the_ultimate_guide/)