As a **fresher**, cracking Google requires a strong grasp of **DSA, problem-solving, CS fundamentals, and projects** to stand out. Since you have **very little time**, here's a **highly focused** plan with the **best resources** to help you get placed quickly.

🔥 1️⃣DSA & Coding (Most Important for Freshers)

📌 **Books:**
✅ **Cracking the Coding Interview (CTCI) – Gayle Laakmann McDowell** (You already ordered!)
✅ **Elements of Programming Interviews (EPI)** – For **FAANG-level** problems.
✅ **Grokking Algorithms** – If you want a **visual approach** to DSA.

📌 **Online Resources:**

- **LeetCode (Google Top 100 + Blind 75 + Hard Problems) – Best for Google!**
- **Striver's SDE Sheet – Step-by-step DSA roadmap.**
- **Neetcode.io – Best structured LeetCode solutions.**
- **Codeforces (Div 2 + Div 1)** – If you want to **boost problem-solving speed.**

🎯 **Google-Focused DSA Topics:**
✅ Arrays, Strings, Linked Lists, Stacks, Queues
✅ Trees (BST, Trie), Graphs (DFS, BFS, Dijkstra)
✅ Dynamic Programming (Knapsack, LIS, Matrix DP)
✅ Sliding Window, Two Pointers, Bit Manipulation

🚀 2️⃣System Design (Not a Priority for Freshers, But Good to Know)

📌 **Books:**
✅ **Grokking the System Design Interview** – For **Google's high-level design questions**.

📌 **Resources:**

- **System Design Primer (GitHub Repo) – Best free resource!**
- **Backend Masterclass by Hussein Nasser (YouTube)**

🎯 **Topics (Google-Focused for Freshers):**
✅ Load Balancing, Caching (Redis, CDN)

✅ SQL vs NoSQL, Database Indexing
✅ API Rate Limiting, Scalability Basics

🧠 3️⃣ CS Fundamentals (Must-Know for Google)

📌 **Books:**
✅ **Operating Systems: Three Easy Pieces** (OS)
✅ **Computer Networking – Kurose & Ross**
✅ **Database System Concepts – Korth**

📌 **Resources:**

- **CS50 (Harvard's Free Course)** – Covers **OS, DBMS, and Networking**.
- **MIT Missing Semester – Shell scripting, Git, productivity tools**.

🎯 **Google-Focused Topics:**
✅ **OS** – Process Management, Threads, Deadlocks, Virtual Memory
✅ **DBMS** – ACID, Transactions, Indexing, Normalization
✅ **Networking** – TCP/IP, HTTP, Load Balancing, CDN

🤖 4️⃣ ML & Java (Since You're Interested in ML)

📌 **Books:**
✅ **Hands-On Machine Learning with Scikit-Learn & TensorFlow – Aurelien Geron**
✅ **Machine Learning in Java – Bostjan Kaluza**

📌 **Resources:**

- **Google ML Crash Course (FREE!)**
- **fast.ai Deep Learning Course**
- **Stanford CS229 (Andrew Ng's ML Course)**

🎯 **Must-Know Topics:**
✅ Supervised vs. Unsupervised Learning
✅ Neural Networks (CNN, RNN, Transformers)
✅ Feature Engineering, Hyperparameter Tuning
✅ Big Data (Google Cloud AI, TensorFlow, JAX)

💻 5️⃣ Projects (To Make Your Resume Stand Out)

📌 **If You Want to Stand Out as a Fresher:**

✅ **DSA & Problem-Solving Projects** – Competitive Coding & Google Kick Start Participation.

✅ **ML-Based Projects in Java** – AI Chatbot, Stock Price Prediction, Recommendation System.

✅ **Web Development + ML (For Google AI/ML Roles)** – Use Flask, Spring Boot with ML models.

✅ **System Design Mini-Projects** – URL Shortener, Rate Limiter, Tiny URL Clone.

🎯 6️⃣ Mock Interviews & Google's Hiring Process

📌 **Practice Platforms:**

- **LeetCode Mock Interviews (Premium)**
- **Pramp (Peer Mock Interviews, FREE)**
- **Google's Kick Start & Code Jam** (Google's own coding contests)

📌 **Google's Hiring Process for Freshers:**

1️⃣ **Online Assessment (OA)** – DSA Problems on **LeetCode (Google Tag)**.

2️⃣ **Technical Interviews (2-3 Rounds)** – DSA + System Design Basics + CS Fundamentals.

3️⃣ **Behavioral Round** – STAR Method for **Googleyness & Leadership**.

📅 7️⃣ Study Plan (Since You Have Less Time)

🎯 **4-6 Hours Daily Plan:**

✅ **2 Hours – DSA** (LeetCode Hard + Google Top Questions)

✅ **1 Hour – CS Fundamentals** (OS, DBMS, Networks Quick Revisions)

✅ **1 Hour – ML (If ML Role)**

✅ **1 Hour – Mock Interviews (LeetCode/Pramp)**

✅ **30 Mins – Competitive Programming (Codeforces/Kick Start)**

📌 **Final Answer:**

- **For Coding Rounds: CTCI + LeetCode Google Top 100 + Codeforces**

- **For System Design (If Needed): Grokking the SDI + System Design Primer**
- **For CS Concepts: OS (Tanenbaum), DBMS (Korth), Networks (Ross)**
- **For ML Role: Hands-On ML + Google ML Crash Course**
- **For Mock Interviews: Pramp + Google Kick Start + LeetCode Mock Interviews**

Would you like a **custom 30-day roadmap** for **Google Prep** as a fresher? 🚀

Cracking Google as a "freaker" (top-tier candidate) requires a highly strategic approach. Since you have very little time left, I'll break it down into must-have materials and resources for DSA, System Design, ML (for ML roles), and CS fundamentals.

🔥 1️⃣Data Structures & Algorithms (DSA) – Google-Level Prep

📌 Books:

✅ Cracking the Coding Interview (CTCI) – Already ordered! Master it.
✅ Elements of Programming Interviews (EPI) – Tougher than CTCI, good for FAANG-level prep.
✅ Grokking Algorithms – For visual understanding if concepts aren't clear.

📌 Online Resources:

- LeetCode (Google Top 100, Hard Problems) – 🏆 Most Important!
- Neetcode.io (DSA Roadmap + Solutions) – Structured learning.
- Striver's SDE Sheet – Covers all Google-level coding questions.
- Codeforces (Div 2 + Div 1) – For contest-style problem-solving.

🎯 Focus Topics for Google:
✅ Arrays, Strings, Linked List, Stack/Queue, Heap
✅ Trees (BST, Trie), Graphs (DFS, BFS, Dijkstra)
✅ Dynamic Programming (LIS, Knapsack, Matrix DP)
✅ Bit Manipulation, Sliding Window, Two Pointers

🚀 2️⃣System Design (For Senior Roles)

📌 Books:
✅ Grokking the System Design Interview (Best for Beginners)
✅ Designing Data-Intensive Applications – Martin Kleppmann (Advanced)

📌 Resources:

- Backend Masterclass by Hussein Nasser (YouTube) – Free & high quality.
- System Design Primer (GitHub) – Google-level design questions.

- Excalidraw (Tool for System Design Diagrams)

🎯 Focus Topics:
✅ Scalability (Load Balancing, Sharding, Replication)
✅ Databases (SQL vs NoSQL, Indexing, Partitioning)
✅ Caching (Redis, Memcached)
✅ Message Queues (Kafka, RabbitMQ)
✅ CAP Theorem, Consistency, Availability

🧠 3️⃣CS Fundamentals (Asked in Google Interviews)

📌 Books:
✅ Operating Systems: Three Easy Pieces – OS
✅ Computer Networking – Kurose & Ross – Networks
✅ Database System Concepts – Korth – DBMS

📌 Resources:

- CS50 (Harvard's Free Course) – OS, Networks, DBMS.
- GFG for Quick Revisions – OS, DBMS, Networking Cheat Sheets.
- MIT Missing Semester – Shell scripting, Git, productivity tools.

🎯 Topics: ✅ OS (Processes, Threads, Deadlocks, Paging)
✅ DBMS (Indexes, Transactions, ACID, Normalization)
✅ Networking (TCP/IP, HTTP, Load Balancing, CDN)

🤖 4️⃣Machine Learning (For ML Engineer Roles at Google)

📌 Books:
✅ Hands-On Machine Learning with Scikit-Learn & TensorFlow – Aurelien Geron
✅ Pattern Recognition and Machine Learning – Christopher Bishop (Advanced)
✅ Deep Learning – Ian Goodfellow (For hardcore DL roles)

📌 Online Resources:

- Google Machine Learning Crash Course (FREE)
- fast.ai Deep Learning Course
- Andrej Karpathy's Blog (For ML Interviews)

- Stanford CS229 (Machine Learning by Andrew Ng)

🎯 **Must-Know Topics:**
✅ ML Algorithms (Linear Regression, SVM, Decision Trees, KNN)
✅ Deep Learning (CNNs, RNNs, Transformers, Attention)
✅ Feature Engineering & Model Optimization
✅ Big Data & Cloud ML (Google Cloud AI, TensorFlow, JAX)

💻 5️⃣Mock Interviews & Competitive Coding

📌 **Practice Platforms:**

- LeetCode (Mock Interviews Section)
- InterviewBit (Timed Tests)
- Pramp (Peer Mock Interviews, FREE)
- Google's Kick Start & Code Jam (Google's own contests)

📌 **Important Google Past Interview Questions:**

- Blind 75 + Google Top 50 Questions on LeetCode
- Actual Google Interview Repo (GitHub: @jwasham)
- HiredInTech (Behavioral Questions Practice)

🗓️ 6️⃣Study Plan (Since You Have Less Time)

🎯 **4-6 Hours Daily Plan:**
✅ 2 Hours – DSA (LeetCode Hard + Google Top Questions)
✅ 1 Hour – System Design (Grokking the SDI + Practice Q&A)
✅ 30 Mins – CS Fundamentals (Quick Revisions)
✅ 1 Hour – ML (If ML Engineer Role)
✅ 1 Hour – Mock Interviews & Google Contest Problems

Final Answer:

- For Coding Rounds: CTCI + LeetCode Google Top 100 + Codeforces
- For System Design: Grokking SDI + Designing Data-Intensive Apps
- For CS Concepts: OS (Tanenbaum/Kurose), DBMS (Korth), Networks (Ross)

- **For ML Role: Hands-On ML + Google ML Crash Course + TensorFlow/JAX**

Would you like a detailed 30-day roadmap for Google prep? 🚀