

Complete Coding job guide 2023






☰ Tags

Introduction

Are you looking for a comprehensive coding job guide that covers everything from entry-level positions to senior developer roles? Look no further! Our coding job guide has got you covered.

Whether you're a new graduate just starting your career or an experienced developer looking for new opportunities, this guide will provide you with valuable insights and practical tips to help you land your dream job in the tech industry.


Here are some of the key features of our coding job guide:



-  Step-by-step guidance for job seekers at all levels
-  Insider tips on how to stand out in the competitive job market
-  Strategies for networking and building professional relationships
-  Advice on resume writing, interview preparation, and salary negotiation
-  Insights into industry trends and job growth opportunities

With our coding job guide, you'll have all the tools and resources you need to succeed in your job search. So what are you waiting for? Let's get started!

Topics Covered

Our guide covers many important topics related to coding including:

- Internships 
 - Tips on how to find and apply for internships
 - Advice on how to make the most of your internship experience

- Data Structures and Algorithms 
 - An overview of these essential concepts
 - Resources for further learning
- Projects 
 - Ideas for projects to build your portfolio and stand out in the job market



Linkedin profile

☰ Tags

Checklist for an Awesome LinkedIn Profile

Professional Profile Photo

Headline

Summary

Experience and Projects

Checklist for an Awesome LinkedIn Profile

Having a strong LinkedIn profile is essential for professional networking and career growth. Here's a checklist of all the things you need to create an awesome LinkedIn profile:

1. **Professional profile photo:** Use a high-quality headshot that portrays you in a professional manner. Dress appropriately, and make sure the background is simple and uncluttered.
2. **Headline:** This is a brief statement that appears below your name on your profile. It should be a clear and concise summary of your professional abilities, experience and interests.
3. **Summary:** Your summary is an opportunity to showcase your skills, experience, and unique value proposition. It should be well-written, engaging, and highlight your strengths.
4. **Experience:** List all your relevant work experience, starting with your most recent job. Use bullet points to describe your responsibilities and achievements in each role.
5. **Education:** Include all your educational qualifications, including degrees, certifications, and courses.

6. **Skills and endorsements:** Add relevant skills to your profile, and ask your connections to endorse you for those skills.
7. **Recommendations:** Request recommendations from your colleagues, managers, or clients who can speak to your professional abilities and skills.
8. **Interests and hobbies:** Share your interests and hobbies, as they can help you connect with others who have similar passions.
9. **Engagement:** Engage with your connections by commenting, sharing, and liking their posts. This can help you expand your network and build relationships.

By following this checklist, you can create an awesome LinkedIn profile that will help you stand out in the professional world.

Professional Profile Photo

Your professional profile photo is the first impression you make on LinkedIn, so it's important to choose a photo that represents you in the best possible light. Your photo should be a high-quality headshot that's well-lit and portrays you in a professional manner. Make sure you dress appropriately for your industry, and choose a simple, uncluttered background that won't distract from your face. Avoid using selfies or casual photos, and make sure your photo is recent and looks like you. Remember, your photo is an important part of your personal brand, so take the time to get it right.

Headline

Your headline is a brief statement that appears below your name on your LinkedIn profile. It should be a clear and concise summary of your professional abilities, experience, and interests. Use keywords that are relevant to your industry and job search, and highlight your unique value proposition.

Summary

Your summary is an opportunity to showcase your skills, experience, and unique value proposition. It should be well-written, engaging, and highlight your strengths. Use a conversational tone and avoid using jargon or buzzwords. Be specific about your accomplishments and the value you can bring to potential employers. Keep in mind that

your summary is often the first thing a recruiter or hiring manager sees, so make it count.

Experience and Projects

When listing your work experience on LinkedIn, make sure to include all relevant jobs, starting with your most recent position. Use bullet points to describe your responsibilities and achievements in each role, highlighting the skills and experience you gained. If you've worked on any projects, be sure to include those as well. Projects can show potential employers that you have a diverse skill set and are capable of working on a variety of tasks. Provide a brief description of each project and explain your role in it. Don't be afraid to highlight any challenges you faced and how you overcame them. By showcasing your experience and projects, you can demonstrate your value to potential employers and stand out in the professional world.



Cold emailing

☰ Tags

[Introduction](#)

[Referrals](#)

[Examples](#)

[Example 1](#)

[Example 2](#)

[Example 3](#)

[Bad example](#)

[Good example](#)

Introduction

Cold email means an email without specifically knowing the person who's receiving it. This might sound absurd, but works a **lot of time**. So it's like sending emails or messages on LinkedIn to your connections at Google. Will they reply? Maybe not, but it's likely that they see the email and forward it to the hiring manager (if they like your profile).

One thing which makes a good cold email/message is a selling point. Imagine yourself as the recruiter or the software engineer at a company and getting hundreds if not thousands of emails from different students. How would you differentiate a normal candidate from a special one?

Here is a template for a perfect cold email:

(ROLE): Name -> Eg: Software Engineering Intern Candidate: Arya...
 _ * x

Recipients

(ROLE): Name -> Eg: Software Engineering Intern Candidate: Aryan Singh

Hello {name},

Introduction - 1 liner
 Hope you're doing well. I'm a computer science student at the {University name}, and am really interested in the {opportunity} at {Company}.

Past experience or Projects - 2-3 lines
 This past summer, I was a {position} at {company}, a {explain what the company does}. I also work as {other work experience (current)}, if working for startups - mention the level, funding, role). I've (won/participated) various hackathons in the past, with projects like {project 1, project 2, project 3}. Add an explainer comment with each project.

Selling Point - GitHub, LinkedIn profile / Some other achievements
 You can check out my 50+ open source projects on [GitHub](#).

Sweet Ending
 I've attached my resume for your reference. Please, let me know if you have any questions. I look forward to your response.

Signature
 Regards,
 {Name}
 {Major}
 {University}

↶ ↷
Sans Serif
TT
B
I
U
A
☰
☷
☰
☷
☰
☷
☰
☷
☰
☷
☰
☷

Send
📎
🔗
😊
📷
📧

⋮
🗑️

Sending connection requests on LinkedIn is also a part of networking which is essential for your process.

Send a personalized message to someone you don't know so that they can read before accepting the request. It's a pleasant way to introduce yourself and you can do it in the following way:



"Hey person, hope you're doing well. It was great talking to you the other day about xyz. I would love to further connect with you on LinkedIn"



"Hey person, hope you're doing well. I loved your Blocktrain.info website and would love to talk to you regarding some contributions. Thank you"

- Start with a clear and concise subject line that explains the purpose of your email
- Personalize the email by addressing the recipient by name, and referencing any common connections or interests

- Keep the email short and to the point, focusing on how you can help the recipient or how your product or service can benefit them
- Be respectful and professional, and avoid being too salesy or aggressive
- Include a call to action that tells the recipient what you would like them to do next, such as scheduling a call or meeting

Overall, the key to a successful cold email is to make it relevant, valuable, and respectful to the recipient.

Referrals

Most of the software companies offer you to apply with a referral of their employee. This trick usually works but lately everyone has been sending out cold emails to ask for referrals. However, it definitely makes you a better candidate than others and can get you the first round interview quicker.

If you know someone from any software company, then just send them a message asking for a referral. They can either send you a specific link for referrals or just forward your resume to the hiring manager - both of which will help get you noticed over other candidates. **Developers working at companies often get incentives if someone joins the company from their referral, so there's nothing to feel shy about in this.**

You can also ask for referrals from LinkedIn or emails using the “cold email” approach, which is discussed below. Pick out the companies you're really passionate about and then devote your time to email them properly, rather than emailing literally everyone on your contact list. People are also more likely to give you a referral if they went to the same school as you. You'll be surprised how well this works out and helps you with your job hunt.

Examples

Example 1

Subject: Introducing My Marketing Agency

Hi [Name],

I hope this email finds you well. My name is [Your Name] and I am the owner of [Company Name], a marketing agency that specializes in helping small businesses increase their online presence.

I came across your business and was impressed by the work you are doing in the [Industry] industry. I believe that with the right marketing strategies, we could help you attract even more customers and drive even more sales.

If you're interested, I'd love to schedule a call to discuss your marketing needs in more detail.

Best regards,
[Your Name]

Example 2

Subject: Collaboration Opportunity for [Your Company]

Hi [Name],

I'm [Your Name], the founder of [Your Company], a [Company Description] company. I came across your work and was really impressed by the quality and creativity of your [Product/Service].

I'm reaching out because I think there may be an opportunity for us to collaborate. At [Your Company], we specialize in [Your Specialty], and I believe that by combining our expertise, we could create something truly unique and valuable for our customers.

If you're interested in exploring this idea further, I'd love to schedule a call to discuss it in more detail.

Best regards,
[Your Name]

Example 3

Subject: Request for a Meeting

Hi [Name],

I hope this email finds you well. My name is [Your Name] and I am a [Your Job Title] at [Your Company]. I came across your work and was really impressed by the innovative approach you are taking in the [Industry] industry.

I would love the opportunity to learn more about your work and see if there are any opportunities for collaboration. Would it be possible to schedule a meeting to discuss this further?

I look forward to hearing from you.

Best regards,

[Your Name]

Bad example

Greetings Aryan,

Hope you are doing well :)

I was wondering if I could get a referral for Google's step internship program, I'll share job id with you

(Step internship is Google's program where they select candidates only from 1st and 2nd year)

I have got a pretty impressive profile and have been into tech since childhood

Please let me know would really appreciate it

- Did not mention resume
- Did not mention skills
- Why will anyone add a referral from this?

Good example

Hope you are doing well! I am a second year Electrical Engineering student at IIT Ropar. I came across your SWE role at Google and am interested in applying for STEP internship opportunity there.

Here's the link:

<https://careers.google.com/jobs/results/136871223167460038-software-student-training-in-engineering-program-step-intern-2023/>

Would you please be open to submitting a referral for me.

Here's the link for my CodeForces profile:

https://codeforces.com/profile/Mehul_Gupta

Here's the link for my CodeChef profile:

https://www.codechef.com/users/mehul_g28

Looking forward to hearing from you.
Thanks



Startups

☰ Tags

[Hiring at Startups vs. Bigger Companies](#)

[Skills vs. DSA or Theoretical Knowledge](#)

[Cultural Fit](#)

[Work-Life Balance](#)

[How to Apply](#)

[How to Cold Email](#)

[How to Network](#)

[Resources](#)

[Work-Life Balance](#)

[Startups hiring](#)

[LinkedIn Strategy to Reach Out to Recruiters](#)

[Optimize Your Profile](#)

[Connect With Recruiters](#)

[Join LinkedIn Groups](#)

[Engage With Company Pages](#)

Hiring at Startups vs. Bigger Companies

When it comes to hiring, startups often have different priorities than larger companies. Here are some key differences to keep in mind:

Skills vs. DSA or Theoretical Knowledge

Startups typically place a greater emphasis on a candidate's skills and experience rather than their ability to solve complex algorithms or demonstrate theoretical knowledge. They need people who can hit the ground running and start making an impact right away.

Cultural Fit

Startups are often more focused on building a strong team culture. They want to ensure that everyone is aligned on the company's mission and values and can work collaboratively towards a common goal.

Work-Life Balance

At early stage startups, work-life balance can be harder to achieve. With limited resources and tight deadlines, employees may need to work longer hours or be more flexible with their schedules.

How to Apply

If you're interested in working for a startup, there are a few things you can do to increase your chances of getting hired:

- Tailor your resume and cover letter to the specific company and role you're applying for
- Highlight your relevant skills and experience
- Be prepared to talk about your passion for the company's mission and values

How to Cold Email

Cold emailing can be a great way to get your foot in the door at a startup. Here are some tips for crafting an effective email:

- Keep it short and to the point
- Explain why you're interested in the company and what you can bring to the table
- Make it easy for the recipient to respond by including your contact information

How to Network

Networking is key to landing a job at a startup. Here are some ways to expand your network:

- Attend industry events and conferences
- Connect with people on LinkedIn and Twitter
- Join relevant Facebook and Slack groups

Resources

Here are some resources to help you in your startup job search:

- Y Combinator
- Product Hunt
- LinkedIn
- Twitter
- AngelList

When it comes to hiring, startups often have different priorities than larger companies. Here are some key differences to keep in mind:

Startups are often more focused on building a strong team culture. They want to ensure that everyone is aligned on the company's mission and values and can work collaboratively towards a common goal.

Work-Life Balance

At early stage startups, work-life balance can be harder to achieve. With limited resources and tight deadlines, employees may need to work longer hours or be more flexible with their schedules.

Startups hiring

The following list is not exhaustive and is subject to change. These are some of the top tech startups that hire software engineers:

1. Airbnb
2. Amazon
3. Asana
4. Atlassian
5. Automattic
6. Betterment

7. Brex
8. Coinbase
9. Dropbox
10. GitLab
11. Instacart
12. Lyft
13. MongoDB
14. Palantir
15. Peloton
16. Postmates
17. Robinhood
18. Slack
19. Stripe
20. Uber
21. WeWork
22. Zoom
23. Square
24. Twilio
25. Snowflake

LinkedIn Strategy to Reach Out to Recruiters

If you're looking to get hired by a startup, LinkedIn can be a powerful tool to connect with recruiters and hiring managers. Here's how you can use LinkedIn to increase your chances of landing a job:

Optimize Your Profile

Your LinkedIn profile is your online resume, so it's important to make sure it's up-to-date and showcases your skills and experience. Here are some tips for optimizing your profile:

- Use a professional-looking headshot
- Write a compelling headline that highlights your skills and experience
- Customize your LinkedIn URL
- Write a summary that showcases your accomplishments and career goals
- List your relevant skills and experiences

Connect With Recruiters

Once your profile is optimized, it's time to start connecting with recruiters. Here's how you can find and connect with recruiters on LinkedIn:

- Use LinkedIn's Advanced Search feature to find recruiters who work at companies you're interested in
- Send recruiters a personalized connection request that highlights your skills and experience
- Follow up with recruiters after connecting to express your interest in working for their company

Join LinkedIn Groups

LinkedIn Groups can be a great way to connect with people in your industry and get noticed by recruiters. Here are some tips for using LinkedIn Groups to your advantage:

- Join groups related to your industry or the companies you're interested in
- Participate in group discussions and share your expertise
- Connect with group members who work at companies you're interested in

Engage With Company Pages

Following company pages on LinkedIn can give you insight into a company's culture and hiring needs. Here's how you can engage with company pages:

- Follow the company pages of the startups you're interested in

- Like and comment on company posts to show your engagement
- Set up job alerts to get notified when a company posts a new job opening

By following these LinkedIn strategies, you can increase your chances of getting noticed by recruiters and landing your dream job at a startup.



Coding projects

☰ Tags

[General Ideas](#)

[Clone projects](#)

[Web3 projects](#)

[Frontend projects](#)

[Backend projects](#)

General Ideas

Project Idea	Difficulty	Programming Language
Simple Calculator	Easy	Python, JavaScript, Java, C++
To-do List	Easy	Python, JavaScript, Java, C++
Weather App	Easy	Python, JavaScript
Hangman Game	Easy	Python, JavaScript
Tic Tac Toe Game	Easy	Python, JavaScript
Currency Converter	Easy	Python, JavaScript
Simple Chat Application	Medium	Python, JavaScript, Java
Blogging Platform	Medium	Python, JavaScript, Ruby on Rails
E-commerce Website	Medium	Python, JavaScript, Ruby on Rails
Online Voting System	Medium	Python, Java
Social Media Platform	Hard	Python, JavaScript, Ruby on Rails

Clone projects

App Clone	Description	Difficulty	Programming Language
Instagram Clone	A social media app for sharing photos	Hard	React, Node.js,

	and videos		MongoDB
Facebook Clone	A social media platform for connecting with friends and family	Hard	React, Node.js, MongoDB
Twitter Clone	A micro-blogging platform for sharing short messages	Medium	React, Node.js, MongoDB
Uber Clone	A ride-sharing app for connecting riders with drivers	Hard	React Native, Node.js, MongoDB
Airbnb Clone	A platform for renting homes and apartments from local hosts	Hard	React, Node.js, MongoDB
TikTok Clone	A short-form video sharing app	Hard	React Native, Node.js, MongoDB
WhatsApp Clone	A messaging app for sending text and multimedia messages	Medium	React Native, Node.js, MongoDB
Reddit Clone	A platform for sharing news and content with communities	Hard	React, Node.js, MongoDB
LinkedIn Clone	A professional networking platform	Hard	React, Node.js, MongoDB
YouTube Clone	A video sharing platform	Hard	React, Node.js, MongoDB

Web3 projects

Project	Description	Tech Stack
Decentralized Exchange (DEX)	A platform for trading cryptocurrencies in a decentralized manner	Solidity, Web3.js, React, Node.js, Ethereum
Non-Fungible Tokens (NFTs) Marketplace	A platform for buying, selling, and trading unique digital assets	Solidity, Web3.js, React, Node.js, Ethereum
Decentralized Finance (DeFi) Application	A suite of financial applications that operate on a decentralized blockchain	Solidity, Web3.js, React, Node.js, Ethereum
Cryptocurrency Wallet	A digital wallet for storing and managing various	React Native,

	cryptocurrencies	Node.js, Web3.js, Ethereum
DAO (Decentralized Autonomous Organization)	A community-driven organization that operates using smart contracts on a blockchain	Solidity, Web3.js, React, Node.js, Ethereum
Gaming Platform	A platform for creating, playing, and trading blockchain-based games	Solidity, Web3.js, React, Node.js, Ethereum
Supply Chain Management System	A system for tracking the movement of goods through the supply chain using blockchain technology	Solidity, Web3.js, React, Node.js, Ethereum
Identity Verification System	A decentralized system for verifying and authenticating identity using blockchain technology	Solidity, Web3.js, React, Node.js, Ethereum

Frontend projects

Project Idea	Description	Difficulty
To-Do List	A simple app for creating and managing to-do items	Easy
Calculator	A basic calculator app for performing arithmetic operations	Easy
Weather App	An app for displaying weather information based on user input	Medium
Pomodoro Timer	A productivity tool that uses the Pomodoro Technique to manage time	Medium
Recipe App	An app for finding and sharing recipes with others	Medium
E-commerce Website	A website for selling products online with shopping cart and checkout functionality	Hard
Social Media App	An app for sharing content and connecting with other users	Hard
Music Streaming Service	A web-based music streaming platform with user accounts and subscription options	Hard
Travel Planning Tool	An app for planning trips, including flights, accommodations, and activities	Hard
Stock Trading	A platform for trading stocks and monitoring financial data	Hard

Platform		
----------	--	--

Backend projects

Project Idea	Description	Difficulty
URL Shortener	An app for shortening long URLs to make them easier to share	Easy
Contact Form	A form for users to submit contact information that is stored in a database	Easy
Blogging Platform	A web application for creating, managing, and publishing blog posts	Medium
Job Board	A platform for posting job listings and managing job applications	Medium
Chat Application	A messaging app for communicating with other users in real-time	Medium
Social Network	A platform for connecting with friends and sharing content	Hard
E-commerce Platform	A web-based platform for selling products with shopping cart and checkout functionality	Hard
Stock Trading Platform	A platform for trading stocks and monitoring financial data	Hard
Food Ordering System	A system for ordering food from restaurants and tracking delivery	Hard
Cloud Storage Service	A service for storing and accessing files in the cloud	Hard



150 DSA questions

☰ Tags

[Arrays](#)

[Linked list](#)

[Binary search](#)

[Sliding window](#)

[2 pointers](#)

[Stacks, Queues](#)

[BFS, DFS](#)

[Recursion](#)

[Backtracking](#)

[Trees](#)

[Dynamic programming](#)

[Graphs](#)

[Additional questions](#)

[Random](#)

[Videos](#)

Arrays

Easy

- <https://leetcode.com/problems/roman-to-integer/>
- <https://leetcode.com/problems/valid-parentheses/>
- <https://leetcode.com/problems/remove-duplicates-from-sorted-array/>
- <https://leetcode.com/problems/remove-element/>
- <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>
- <https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/>

- <https://leetcode.com/problems/intersection-of-two-arrays-ii/>
- <https://leetcode.com/problems/single-number/>
- <https://leetcode.com/problems/contains-duplicate/>
- <https://leetcode.com/problems/plus-one/>
- <https://leetcode.com/problems/move-zeroes/>
- <https://leetcode.com/problems/rotate-image/>

Medium

- <https://leetcode.com/problems/3sum/>
- <https://leetcode.com/problems/4sum/>
- <https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/>
- <https://leetcode.com/problems/group-anagrams/>
- <https://leetcode.com/problems/reduce-array-size-to-the-half/>
- <https://leetcode.com/problems/merge-intervals/>

Linked list

Easy

- <https://leetcode.com/problems/delete-node-in-a-linked-list/>
- <https://leetcode.com/problems/remove-nth-node-from-end-of-list/>
- <https://leetcode.com/problems/merge-two-sorted-lists/>
- <https://leetcode.com/problems/palindrome-linked-list/>
- <https://leetcode.com/problems/linked-list-cycle/>

Medium

- <https://leetcode.com/problems/intersection-of-two-linked-lists/>
- <https://leetcode.com/problems/remove-linked-list-elements/>
- <https://leetcode.com/problems/middle-of-the-linked-list/>
- <https://leetcode.com/problems/merge-k-sorted-lists/>

Binary search

Easy

- <https://leetcode.com/problems/binary-search/>
- <https://leetcode.com/problems/intersection-of-two-arrays/>
- <https://leetcode.com/problems/first-bad-version/>
- <https://leetcode.com/problems/arranging-coins/>
- <https://leetcode.com/problems/search-insert-position/>

Medium

- <https://leetcode.com/problems/search-in-rotated-sorted-array/>
- <https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/>
- <https://leetcode.com/problems/kth-smallest-element-in-a-bst/>
- <https://leetcode.com/problems/find-peak-element/>
- <https://leetcode.com/problems/split-array-largest-sum/>

Sliding window

Read

- [Leetcode Pattern 2 | Sliding Windows for Strings | by csgator | Leetcode Patterns](#)

Easy/Medium

- <https://leetcode.com/problems/longest-substring-without-repeating-characters/>
- <https://leetcode.com/problems/find-all-anagrams-in-a-string/description/>
- <https://leetcode.com/problems/minimum-window-substring/description/>
- <https://leetcode.com/problems/count-number-of-nice-subarrays/>
- <https://leetcode.com/problems/fruit-into-baskets/>

2 pointers

- <https://leetcode.com/problems/intersection-of-two-arrays/>

- <https://leetcode.com/problems/maximum-ascending-subarray-sum/>
- <https://leetcode.com/problems/backspace-string-compare/>
- <https://leetcode.com/problems/long-pressed-name/>
- <https://leetcode.com/problems/fruit-into-baskets/>
- <https://leetcode.com/problems/max-consecutive-ones-iii/>
- <https://leetcode.com/problems/container-with-most-water/>

Stacks, Queues

- Easy
- <https://leetcode.com/problems/valid-parentheses/>
- <https://leetcode.com/problems/implement-queue-using-stacks/>
- <https://leetcode.com/problems/min-stack/>
- Medium
- <https://leetcode.com/problems/design-circular-queue/>
- <https://leetcode.com/problems/decode-string/>
- <https://leetcode.com/problems/open-the-lock/>
- <https://leetcode.com/problems/daily-temperatures/>
- <https://leetcode.com/problems/minimum-add-to-make-parentheses-valid/>

BFS, DFS

Read

- [Leetcode Pattern 1 | BFS + DFS == 25% of the problems — part 1](#)
- [Leetcode Pattern 1 | DFS + BFS == 25% of the problems — part 2](#)

Questions

- <https://leetcode.com/problems/flood-fill/>
- <https://leetcode.com/problems/binary-tree-preorder-traversal/>

- <https://leetcode.com/problems/number-of-islands/>
- <https://leetcode.com/problems/walls-and-gates/>
- <https://leetcode.com/problems/max-area-of-island/>
- <https://leetcode.com/problems/number-of-provinces/>
- <https://leetcode.com/problems/perfect-squares/>
- <https://leetcode.com/problems/course-schedule/>
- <https://www.geeksforgeeks.org/detect-cycle-undirected-graph/>
- <https://leetcode.com/problems/word-ladder/>
- <https://leetcode.com/problems/01-matrix/>
- <https://leetcode.com/problems/rotting-oranges/>
- <https://leetcode.com/problems/perfect-squares/>
- <https://leetcode.com/problems/all-paths-from-source-to-target/>
- <https://leetcode.com/problems/number-of-closed-islands/>

Recursion

Easy

- [509. Fibonacci Number](#)
- [Reverse String](#)
- [24. Swap Nodes in Pairs](#)
- [206. Reverse Linked List](#)
- [Leetcode #700 Search in a Binary Search Tree](#)
- [70. Climbing Stairs](#)
- [Leetcode #50 Pow\(x, n\)](#)

Backtracking

Read

- [Leetcode Pattern 3 | Backtracking | by csgator | Leetcode Patterns](#)

- A general approach to backtracking questions in Java (Subsets, Permutations, Combination Sum, Palindrome Partitioning)

Easy

- Word Search
- Leetcode #78 Subsets
- 90. Subsets II
- Letter Case Permutation

Medium

- 39. Combination Sum
- 17. Letter Combinations of a Phone Number
- Combinations
- Leetcode : Combination Sum II
- 216. Combination Sum III
- Combination Sum IV
- 46. Permutations
- 47. Permutations II
- 31. Next Permutation
- 51. N-Queens

Trees

Read

- Leetcode Pattern 0 | Iterative traversals on Trees | by csgator | Leetcode Patterns

Easy

- <https://leetcode.com/problems/binary-tree-preorder-traversal/>
- <https://leetcode.com/problems/binary-tree-inorder-traversal/>
- <https://leetcode.com/problems/binary-tree-postorder-traversal/>

- <https://leetcode.com/problems/validate-binary-search-tree/>
- <https://leetcode.com/problems/minimum-distance-between-bst-nodes/>
- <https://leetcode.com/problems/symmetric-tree/>
- <https://leetcode.com/problems/same-tree/>
- <https://leetcode.com/problems/path-sum/>
- <https://leetcode.com/problems/maximum-depth-of-binary-tree/>
- <https://leetcode.com/problems/convert-sorted-array-to-binary-search-tree/>
- Medium
- <https://leetcode.com/problems/validate-binary-search-tree/>
- <https://leetcode.com/problems/binary-search-tree-iterator/>
- <https://leetcode.com/problems/unique-binary-search-trees/>
- <https://leetcode.com/problems/serialize-and-deserialize-bst/>
- <https://leetcode.com/problems/binary-tree-right-side-view/>
- <https://leetcode.com/problems/binary-tree-level-order-traversal/>
- <https://leetcode.com/problems/binary-tree-level-order-traversal-ii/>
- <https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal/>

Dynamic programming

Easy

- <https://leetcode.com/problems/maximum-subarray/>
- <https://leetcode.com/problems/fibonacci-number/>
- <https://leetcode.com/problems/climbing-stairs/>
- <https://leetcode.com/problems/min-cost-climbing-stairs/>
- <https://leetcode.com/problems/n-th-tribonacci-number/>

Medium

- <https://leetcode.com/problems/coin-change/>

- <https://leetcode.com/problems/minimum-falling-path-sum/>
- <https://leetcode.com/problems/minimum-cost-for-tickets/>
- <https://leetcode.com/problems/2-keys-keyboard/>
- <https://leetcode.com/problems/maximum-product-subarray/>
- <https://leetcode.com/problems/triangle/>
- <https://leetcode.com/problems/ones-and-zeroes/>
- <https://leetcode.com/problems/longest-arithmetic-subsequence/>
- <https://leetcode.com/problems/partition-equal-subset-sum/>
- <https://leetcode.com/problems/house-robber/>
- <https://leetcode.com/problems/decode-ways/>
- <https://leetcode.com/problems/word-break/>
- <https://leetcode.com/problems/edit-distance/>
- <https://leetcode.com/problems/longest-increasing-subsequence/>

Graphs

Easy

- <https://leetcode.com/problems/employee-importance/>
- <https://leetcode.com/problems/find-the-town-judge/>

Medium

- <https://leetcode.com/problems/course-schedule-ii/>
- <https://leetcode.com/problems/redundant-connection/>
- <https://leetcode.com/problems/surrounded-regions/>
- <https://leetcode.com/problems/accounts-merge/>
- <https://leetcode.com/problems/network-delay-time/>
- <https://leetcode.com/problems/is-graph-bipartite/>
- <https://leetcode.com/problems/find-eventual-safe-states/>

- <https://leetcode.com/problems/keys-and-rooms/>
- <https://leetcode.com/problems/possible-bipartition/>
- <https://leetcode.com/problems/most-stones-removed-with-same-row-or-column/>
- <https://leetcode.com/problems/rotting-oranges/>
- <https://leetcode.com/problems/number-of-operations-to-make-network-connected/>

Additional questions

- <https://leetcode.com/problems/longest-common-prefix/>
- <https://leetcode.com/problems/implement-trie-prefix-tree/>

Random

- <https://leetcode.com/explore/>

Videos

Some long videos, to revise or study in one long stretch!

- [Data Structures Easy to Advanced Course - Full Tutorial from a Google Engineer](#)
- [Introduction to Data Structures & Algorithms | Course Details & Prerequisites](#)
- [Algorithms Course - Graph Theory Tutorial from a Google Engineer](#)



Competitive programming

☰ Tags

[Introduction](#)

[Studying](#)

[Practice](#)

[Complete Roadmap for Beginners](#)

- [1. Understand the Basics of Programming Languages:](#)
- [2. Choose a Platform:](#)
- [3. Start with Simple Problems:](#)
- [4. Analyze Sample Solutions:](#)
- [5. Participate in Contests:](#)
- [6. Learn Data Structures and Algorithms:](#)
- [7. Practice, Practice, Practice:](#)
- [8. Collaborate and Learn from Others:](#)
- [9. Keep Yourself Updated:](#)
- [10. Don't Give Up \(Most Important\)](#)

[Common topics](#)

[Platforms](#)

Introduction

Competitive programming is the practice of solving algorithmic problems under time constraints, often in a competitive setting such as a programming contest. The goal is to write efficient and correct code that can solve a given problem within the time limits of the contest. Some common topics in competitive programming include:

- Data structures: arrays, linked lists, stacks, queues, trees, graphs, etc.
- Algorithms: sorting, searching, dynamic programming, greedy algorithms, etc.
- Number theory and combinatorics
- Game theory
- Graph algorithms: graph traversals, shortest paths, minimum spanning tree, etc.

- String algorithms: pattern matching, string compression, etc.
- Geometry
- Combinatorial optimization
- Advanced data structures like Trie, Segment Tree, Fenwick tree, etc.
- Computational geometry

Studying

There are several ways to study competitive programming, some of the most effective ways include:

1. **Practicing problems:** The best way to improve at competitive programming is to practice solving problems. There are many online platforms such as CodeForces, Leetcode, HackerRank, CodeChef, etc. where you can find a wide variety of problems to solve.
2. **Participating in contests:** Participating in online programming contests is a great way to get a feel for the time pressure and competition of a real contest. It will also help you to learn from other participants' solutions and approaches.
3. **Reading books and tutorials:** There are many books and tutorials available online that cover the topics and techniques used in competitive programming. Some popular books include "Introduction to Algorithms" by Cormen, "Algorithms Unlocked" by Thomas H. Cormen and "Competitive Programming" by Steven and Felix Halim
4. **Learning from others:** Join online communities or forums where competitive programmers share their tips and solutions. This can help you learn from more experienced programmers and stay up-to-date on the latest techniques and trends.
5. **Implementing algorithms:** It's important to understand the theory behind the algorithms but equally important to be able to implement them. Try to implement the algorithms you learn, it will help you to understand it better and also help with debugging your code.
6. **Regularly revise the topics:** Competitive programming requires a lot of practice and revision. Make sure to revise the concepts and the problems you solved regularly, it will help you to keep the concepts fresh in your mind.

Practice

1. Sorting and searching: questions that involve sorting and searching algorithms, such as finding the kth smallest element in an array or sorting a list of numbers in a specific order.
2. Dynamic programming: questions that involve breaking a problem down into smaller subproblems and storing solutions to those subproblems to avoid redundant computation.
3. Graph algorithms: questions that involve traversing or manipulating graph data structures, such as finding the shortest path between two nodes in a graph or determining the minimum spanning tree of a graph.
4. String algorithms: questions that involve manipulating strings, such as finding the longest common substring or checking if a string is a palindrome.
5. Combinatorial optimization: questions that involve finding the best solution among a large number of possibilities, such as the traveling salesman problem or the knapsack problem.
6. Number theory: questions that involve understanding of number properties or manipulation of numbers.
7. Divide and conquer: questions that involve breaking the problem into smaller subproblems and solve them independently.
8. Greedy algorithm: questions that involve making the locally optimal choice at each stage with the hope of finding the global optimum.
9. Backtracking: questions that involve finding all possible solutions by incrementally building candidate solutions and abandoning a candidate as soon as it is determined that the candidate cannot possibly be completed to a valid solution.
10. Bit manipulation: questions that involve manipulating the individual bits of a number, such as counting the number of set bits in a binary representation of a number.

Complete Roadmap for Beginners

1. Understand the Basics of Programming Languages:

The first step to start Competitive Programming is to have a clear understanding of a programming language. A few popular programming languages for competitive programming are C++, Python, and Java. Once you select your preferred language, start learning the syntax, structure, and basic constructs like loops, conditional statements, and functions.

2. Choose a Platform:

There are several online platforms available for Competitive Programming. Some of the popular ones include Codeforces, CodeChef, HackerRank, and LeetCode. Choose a platform based on your skill level, and join their community. These platforms have multiple rounds of contests, allowing you to participate in regular competitions.

3. Start with Simple Problems:

Once you understand the basics of programming languages, start solving simple problems. These problems are generally divided into categories based on difficulty levels, like easy, medium, and hard. Start with easy problems and gradually move on to harder ones as you gain experience.

4. Analyze Sample Solutions:

After solving a problem, analyze the sample solutions. Try to understand how they have approached the problem and solved it. This will give you an idea of how to approach similar problems in the future.

5. Participate in Contests:

Participating in contests is an excellent way to improve your skills. These contests have a set of problems with varying difficulty levels. Try to solve as many problems as possible within the given time frame. After the contest, analyze your solutions and compare them with others.

6. Learn Data Structures and Algorithms:

Data structures and algorithms are essential concepts in Competitive Programming. They help in solving problems efficiently. Some of the essential data structures and algorithms are arrays, linked lists, stacks, queues, trees, graphs, sorting algorithms, and searching algorithms. Learn these concepts and practice implementing them.

7. Practice, Practice, Practice:

Practice is the key to becoming a successful Competitive Programmer. Spend time daily practicing problems and participating in contests. This will help you identify your weak areas and work on them.

8. Collaborate and Learn from Others:

Collaborating with other Competitive Programmers is an excellent way to learn new concepts and strategies. Join online communities and forums, and discuss your problems and solutions with other programmers. This will help you gain new perspectives and improve your problem-solving skills.

9. Keep Yourself Updated:

Competitive Programming is constantly evolving, and new concepts and techniques emerge regularly. Keep yourself updated with the latest trends and changes in the field. Read blogs, watch videos, and participate in discussions to stay on top of the latest trends.

10. Don't Give Up (Most Important)

Competitive Programming can be challenging and frustrating at times. But don't give up! Keep practicing, and eventually, you will become better. Remember, becoming a successful Competitive Programmer takes time, effort, and patience.

Common topics

- Data structures: arrays, linked lists, stacks, queues, trees, graphs, etc.
- Algorithms: sorting, searching, dynamic programming, greedy algorithms, etc.
- Number theory and combinatorics
- Game theory
- Graph algorithms: graph traversals, shortest paths, minimum spanning tree, etc.
- String algorithms: pattern matching, string compression, etc.
- Geometry

- Combinatorial optimization
- Advanced data structures like Trie, Segment Tree, Fenwick tree, etc.
- Computational geometry

In addition to the topics listed above, some other important areas in competitive programming include:

- Dynamic programming: This involves breaking a problem down into smaller subproblems and storing solutions to those subproblems to avoid redundant computation.
- Divide and conquer: This involves breaking the problem into smaller subproblems and solving them independently.
- Greedy algorithm: This involves making the locally optimal choice at each stage with the hope of finding the global optimum.
- Backtracking: This involves finding all possible solutions by incrementally building candidate solutions and abandoning a candidate as soon as it is determined that the candidate cannot possibly be completed to a valid solution.
- Bit manipulation: This involves manipulating the individual bits of a number, such as counting the number of set bits in a binary representation of a number.

It's important to have a good understanding of these topics if you want to become a successful competitive programmer. You can start by practicing problems related to these topics on online platforms or participating in online programming contests.

Platforms

There are several platforms available for practicing competitive programming, some of the most popular ones being CodeForces, LeetCode, HackerRank, and CodeChef.



Remote job

☰ Tags

[What AI says?](#)

[Websites](#)

[Email people](#)

[Other websites](#)

What AI says?

1. Build your online presence: Make sure your LinkedIn profile is up-to-date and includes any relevant work experience, skills, and achievements. Consider creating a personal website or blog to showcase your work and skills.
2. Network with people in your industry: Attend industry events, join relevant online groups or forums, and reach out to people in your network to see if they know of any remote job opportunities.
3. Search for remote job openings: There are many websites and job boards that specialize in remote job listings, such as FlexJobs, Remote.co, and We Work Remotely. You can also check the websites of companies that interest you to see if they have any remote job openings.
4. Tailor your resume and cover letter: When applying for a remote job, it's important to emphasize your ability to work independently and effectively communicate and collaborate with team members remotely. Make sure to highlight any relevant experience you have with remote work, such as freelance or consulting work.
5. Prepare for the interview: Remote interviews can be conducted over the phone, video call, or in person. Make sure to prepare for the interview just as you would for an in-person interview, including researching the company and the role, and practicing your answers to common interview questions.

Why is it hard?

- Need great skills - the team and founders have to trust you a lot more than a normal employee. Having people in the office forces them to work (kind of).

Websites

- FlexJobs: A paid subscription service that curates a wide range of remote and flexible job openings, including part-time, full-time, and freelance positions.
- Remote.co: A website that lists job openings at companies that offer remote work options, as well as resources and advice for remote workers.
- We Work Remotely: A job board for remote jobs in a variety of industries, including tech, marketing, customer service, and more.
- Remote OK: A job board specifically for tech-related remote jobs.
- LinkedIn: A professional networking website that allows you to search for job openings by location, including remote jobs.
- Indeed: A job search website that allows you to search for jobs by location, including remote jobs.
- Upwork: A freelancing website that allows you to find and apply for short-term or long-term freelance and contract work.

Email people

Cold emailing is the only way to go.

Find people here:

- Twitter, LinkedIn
- Angel list
- Y Combinator

LinkedIn offers some other advantages than the other sites. You can comment on posts by recruiters, hiring managers, and software developers who are looking to hire for their companies. You can also message recruiters, managers, or other service providers about the opportunity they've listed somewhere.

Recruiters and engineering managers share a lot of posts about hiring for their team and it's good to look out for those. Like the post, put in a small comment, and then message them personally asking about the opportunity. A sample message would look like this:



“Hey recruiter, hope you’re doing well. I’m really interested in the xyz opportunity you just posted about and would love to talk more about it. I’ve attached my resume and look forward to your response”

AngelList is the most popular website for applying to **startups**. If you feel like a growing company might benefit you the most right now, angelList is the way to go. There's risk involved with all kinds of situations but keeping a long term goal will help you rise well.

Other websites

1. LinkedIn (<https://www.linkedin.com/jobs/>)
2. Monster (<https://www.monster.com/>)
3. Indeed (<https://www.indeed.com/>)
4. Jumpstart (<https://jumpstart.me/forum/all>)
5. Tech internships (<https://www.techinternship.io/>)
6. CS interns (<https://csinterns.com/>)
7. AngelList (<https://angel.co/jobs>)



Mock interview

≡ Tags

Mock Interviews for Job Seekers

Mock interviews are practice interviews designed to simulate real job interviews. They are typically conducted by a friend, mentor, or career counselor and are used to help job seekers prepare for actual job interviews. Participating in mock interviews can have several benefits for job seekers.

Benefits of Mock Interviews

Get familiar with the interview format and structure

Mock interviews allow job seekers to get a sense of what to expect during a real job interview. This can help reduce anxiety and nervousness, as job seekers will be more familiar with the format and structure of the interview.

Practice responses to common interview questions

Mock interviews also allow job seekers to practice their responses to common interview questions. Job seekers can receive feedback on their communication style and body language, which can help them identify and address any weaknesses or areas for improvement in their interview performance.

Receive feedback from a neutral third party

Mock interviews can be a valuable opportunity for job seekers to receive feedback from a neutral third party, such as a mentor or career counselor. This can help job seekers to identify areas of their resume or job application that may be lacking or to address any weaknesses in their interview skills.

Overall, mock interviews can be an effective way for job seekers to improve their chances of success in real job interviews and to increase their confidence and

preparedness.

Check out some resources for finding mock interviews:

- [Pramp](#)
- [Interviewing.io](#)
- [AlgoExpert](#)
- [TechMockInterview](#)

Tips for Acing Mock Interviews

1. Research the company and the position you are applying for
2. Practice common interview questions and prepare responses
3. Dress appropriately and arrive early
4. Make eye contact and practice good body language
5. Emphasize your strengths and highlight your skills and experiences
6. Be honest and authentic in your responses
7. Ask thoughtful questions about the company and the position
8. Follow up with a thank-you note or email after the interview

By following these tips, you can make the most out of your mock interview experience and be better prepared for actual job interviews.