

Awesome tech job roadmap

Resume, application process, DSA, and awesome Full stack projects

Watch the videos

https://youtube.com/playlist?list=PLxbAS7NVaSZIgqpVNyOA0_ftJN9i1Mls_&feature=shared

{ 30 }

SWE @ GOOGLE

WHY TO JOIN?

REACT and NEXT JS

AI, BLOCKCHAIN, WEB3

OPEN SOURCE PROJECTS

FULL STACK PRODUCTS

What's New?

1:1 MEETINGS AND EVENTS

24/7 COMMUNITY

WHATSAPP + DISCORD

WEEKLY LIVE SESSIONS

Join the 24/7 community - 30DaysCoding.com

1. Ask questions anytime
2. Live events every week
3. Exclusive projects and live classes
4. 1:1 mentorship sessions

Project 16: No code app - internship list
Aryan Singh • 7 views • 1 month ago

Project 17: Streamlit AI app
Aryan Singh • 5 views • 1 month ago

Project 18: Vercel, Next JS templates
Aryan Singh • 4 views • 1 month ago

Project 19: Indie hackers overview
Aryan Singh • 2 views • 4 weeks ago

Project 20: Twitter for developers
Aryan Singh • 8 views • 4 weeks ago

Project 21: Coding roadmaps
Aryan Singh • 27 views • 3 weeks ago

Project 22: Deploying your apps
Aryan Singh • 17 views • 3 weeks ago

Project 23 Figma landing pages
Aryan Singh • 3 views • 3 weeks ago

Resume and application process

Resume, cold email, referrals, job boards, remote jobs

Jake Ryan

123-456-7890 | jake@tsu.edu | linkedin.com/in/jake | github.com/jake

EDUCATION

Southwestern University	Georgetown, TX
<i>Bachelor of Arts in Computer Science, Minor in Business</i>	<i>Aug. 2018 – May 2021</i>
Blinn College	Bryan, TX
<i>Associate's in Liberal Arts</i>	<i>Aug. 2014 – May 2018</i>

EXPERIENCE

Undergraduate Research Assistant

Texas A&M University

- Developed a REST API using FastAPI and PostgreSQL to store data from learning management systems
- Developed a full-stack web application using Flask, React, PostgreSQL and Docker to analyze GitHub data
- Explored ways to visualize GitHub collaboration in a classroom setting

Information Technology Support Specialist

Southwestern University

- Communicate with managers to set up campus computers used on campus
- Assess and troubleshoot computer problems brought by students, faculty and staff
- Maintain upkeep of computers, classroom equipment, and 200 printers across campus

Artificial Intelligence Research Assistant

Southwestern University

May 2019 – July 2019
Georgetown, TX

- Explored methods to generate video game dungeons based off of *The Legend of Zelda*
- Developed a game in Java to test the generated dungeons
- Contributed 50K+ lines of code to an established codebase via Git
- Conducted a human subject study to determine which video game dungeon generation technique is enjoyable
- Wrote an 8-page paper and gave multiple presentations on-campus
- Presented virtually to the World Conference on Computational Intelligence

PROJECTS

Gitlytics | Python, Flask, React, PostgreSQL, Docker

June 2020 – Present

- Developed a full-stack web application using Flask serving a REST API with React as the frontend
- Implemented GitHub OAuth to get data from user's repositories
- Visualized GitHub data to show collaboration
- Used Celery and Redis for asynchronous tasks

Simple Paintball | Spigot API, Java, Maven, TravisCI, Git

May 2018 – May 2020

- Developed a Minecraft server plugin to entertain kids during free time for a previous job
- Published plugin to websites gaining 2K+ downloads and an average 4.5/5-star review
- Implemented continuous delivery using TravisCI to build the plugin upon new a release
- Collaborated with Minecraft server administrators to suggest features and get feedback about the plugin

TECHNICAL SKILLS

Languages: Java, Python, C/C++, SQL (Postgres), JavaScript, HTML/CSS, R

Frameworks: React, Node.js, Flask, JUnit, WordPress, Material-UI, FastAPI

Developer Tools: Git, Docker, TravisCI, Google Cloud Platform, VS Code, Visual Studio, PyCharm, IntelliJ, Eclipse

Libraries: pandas, NumPy, Matplotlib

Jake ryan's resume template

- Use this template
- <https://www.overleaf.com/latex/template/s/jakes-resume/syzfjbzwjncs>
- 4 sections
- 3-4 points
- Live links
- Github links
- NO certificates
- No bullshit
- Nothing older than college unless super important
- DON'T JUST FILL in
- Keep it relevant

Sponsored



MyPerfectResume

<https://www.myperfectresume.com> ::

Free Resume Templates - Fastest Resume Builder Online

Our Automatic **Resume** Templates Make You a Perfect **Resume** in Just a Few Clicks. Free & Easy. Are You Making These Common **Resume** Mistakes? Use Our Free, Automatic...

Free Online Resume Maker

Build a Resume in Minutes. Quick & Easy Resumes w/ Our Templates.

Free Perfect Resume Maker

Build Your Perfect Resume for Free! Choose from Hundreds of Designs.

100+ Resume Formats

Find Expert Resume Formats & Layout Make the Perfect Resume in Minutes.

Sponsored



Resume Now

<https://www.resume-now.com/freeresumetemplates> ::

Free Resume Builder

Build Your Perfect **Resume** Today! Free **Resume** Builder Here. Job search toolkit. Live chat.

Sponsored



Resume Genius

<https://www.resumegenius.com> ::

A Resume Like No Other - Build a Resume In 15 Minutes

Don't Undersell Yourself. Tell Your Story with **Resume** Genius & Get Hired Now. Show That...

STOP SPENDING HOURS ON TOOLS

- Keep it simple
- Simple english
- Simple points
- Learn the basics

The harsh reality is that no one is going to read your resume unless you have something very very impressive

Everything you need!

1. Stop looking for AI tools or templates
2. Make a simple resume and start applying
3. You are not getting opportunities because of your projects, not because your resume is “bad”
4. Just use the template and work on things that matter

Projects + DSA = 99% of your job application

So move on from the fucking resume PLEASE.

Cold emailing and networking

Coders, freelancers, indie hackers, sellers, etc.

Software Engineering Intern : Aryan Singh

Inbox x



Aryan Singh

Hey Amy, Hope you're doing well. I'm a computer science student at the University of Massachusetts, Amherst, and am really interested in the SDE internship oppo



Amy.Padgett@dell.com

to me ▾

Thanks so much for reaching out Aryan! I have shared your resume with some of our engineering teams that still have openings in their summer intern programs. A

Regards,

Software Engineering Intern Interest: Aryan Singh

External

Inbox x



Aryan Singh <aryansingh@umass.edu>

to michala.romano ▾

Hey Michala,

Hope you're doing well. I'm a computer science student at the University of Massachusetts, Amherst, and am really interested in the SDE internship opportunity at MathWorks.

This past summer, I was the Software Engineer Intern at Arrow Electronics, a large distributor of electronic components. I also work as a part-time mobile development engineer for a startup with over \$50k in funding. I've won various hackathons in the past, with projects like Safety for Women, Medical Tracker, and VR Classroom for Students with Disabilities.

You can check out my 50+ open source projects on [GitHub](#).

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

Regards,

Aryan Singh

Computer Science

University of Massachusetts Amherst

LinkedIn: <https://www.linkedin.com/in/aryansingh2/>

GitHub: <https://github.com/aryansingh12>

Normal email + emailing when you have another offer

- Use the offer
- Use the details
- Use the tricks
- Fasten the process

Quora Software Engineer Intern : Aryan Singh



Aryan Singh <aryansingh@umass.edu>

to tnanz ▾

Hey Taylor,

Hope you're doing well. I'm a computer science student at the University of Massachusetts, Amherst, and am really interested in the SDE internship opportunity at Quora. I took the online coding assessment some time back and I've attached my report to this email.

This past summer, I was a Software Engineer Intern at Arrow Electronics, a large distributor of electronic components. I helped them develop an information, reporting, and ecommerce based application. I also work as a part-time mobile development engineer for a startup with over \$50k in funding. I've won various hackathons in the past, with projects like Safety for Women, Medical Tracker, and VR Classroom for Students with Disabilities.

Riot Games Intern Coding Test, Summer 2020

Powered by HackerRank

Hello,

Thank you for applying to our Summer 2020 Internship role at Riot Games! Before we review your application contents, we ask that you complete this Coding Test focused on CS fundamental concepts. **NOTE: All intern roles within Software, Systems, Data, Database Engineering require completion of this Hackerrank test for consideration.**

A few things to note:

- The test duration is **120 minutes** long. Please reserve the appropriate amount of time at a comfortable space to complete the challenge
- All inputs will be from STDIN and outputs will go to STDOUT, so we recommend familiarizing yourself with input parsing and standard stream operations
- The "Submit Code & Continue" button in HackerRank may not save your previous input (i.e. If you complete Q3 then press it, then go back to Q3, the code may no longer be there). Therefore, we encourage you to save your answers in an external code editor as you complete each question in the test
- We recommend completing the challenge within **7 days** of receipt. If you are unable to take the test within this time frame due to reasonable circumstances, please email us so we can accommodate.

If you experience any unexpected issues, please respond to this email.

Fistbumps,

Riot Games

- Heading
- Description
- Test details
- Test results
- Aim of the email
- Result of the email
- Keep it simple
- Provide value
- Concise

Google SDE Internship : Aryan Singh

Inbox x



Aryan Singh <aryansingh@umass.edu>

to bethandre ▾

Wed, Dec 18, 2019, 8:00 AM



Hey Beth,

Hope you're doing well. I took the google coding challenge a couple weeks back and am waiting to hear back. It went really well and am looking forward to the next steps for the internship. I have a competing offer from another company with their deadline coming soon, but I'm still very interested in the SDE intern position with Google.

Please let me know if the interview can be scheduled in the coming couple of weeks. Looking forward to hearing from you. Thank you.

Regards,

Aryan Singh

Computer Science

University of Massachusetts Amherst

[LinkedIn](#), [GitHub](#)

Wed, Dec 18, 2019, 8:07 AM



Thanks for reaching out and letting me know about your offer. Congratulations! The team is working as quickly as they can to review all applications however they may not be able to accommodate your upcoming offer deadline. If you are still interested in pursuing this role I would encourage you to ask for an offer extension. Otherwise, I would be happy to withdraw your application from the process. Please note that a withdraw has no effect on future opportunities. Either way please let me know how you would like to proceed.

Additionally, I sent a message to your recruiter asking that they look at your materials.

Finally, what is your current offer deadline?

Looking forward to hearing from you soon.

Best,
Beth

<input type="checkbox"/>	 Handshake	Inbox	Your application to State Street Corporation was successfully submitted - Your application is confirmed Nice work! You successfully a...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Fidelity Investments was successfully submitted - Your application is confirmed Nice work! You successfully applic...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to iRobot was successfully submitted - Your application is confirmed Nice work! You successfully applied to Compute...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to TJX Companies was successfully submitted - Your application is confirmed Nice work! You successfully applied to ...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Kronos was successfully submitted - Your application is confirmed Nice work! You successfully applied to Summer...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Systems & Technology Research was successfully submitted - Your application is confirmed Nice work! You succe...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to HPR (Hyannis Port Research) was successfully submitted - Your application is confirmed Nice work! You successfu...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Charles River Development was successfully submitted - Your application is confirmed Nice work! You successfull...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Optum, a UnitedHealth Group Company was successfully submitted - Your application is confirmed Nic... + ⚡ 🗑️ 📧 ⏰	10/1/18
<input type="checkbox"/>	 Wayfair Recruiting .	Inbox	Your Wayfair application for Software Engineering Intern - Summer 2019 - Dear ARYAN, Thank you for your interest in a career at Wayf...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Amazon was successfully submitted - Your application is confirmed Nice work! You successfully applied to Amazo...	10/1/18
<input type="checkbox"/>	 Handshake	Inbox	Your application to Handshake was successfully submitted - Your application is confirmed Nice work! You successfully applied to Soft...	10/1/18
<input type="checkbox"/>	 Handshake 3	Inbox	Your application to Alibaba Group was successfully submitted - Your application is confirmed Nice work! You successfully applied to I...	10/1/18

Using a software tool like indeed, handshake, etc is NOT A GOOD PRACTICE.

- Look at this example
- 10000s of people are applying this way
- Very slow and outdated

<input type="checkbox"/>	★	snapshot-emails	Inbox	Next steps for your Google application - Dear Aryan Singh, Thanks again for applying for the role of Software Engineering Intern, BS, ...	11/25/19
<input type="checkbox"/>	★	Jacqueline, me 3	Inbox	Workhuman; Telephone Interview Invitation - Hi Aryan, Thank you so much for your response. At this time, I've scheduled for us to co...	11/25/19
<input type="checkbox"/>	★	Vestmark via Codili.	Inbox	Vestmark invites you to a test at Codility - Hi Aryan Singh, Thank you for your interest in Vestmark! We would like you to take a short p...	11/19/19
<input type="checkbox"/>	★	workday arrow	Inbox	Thanks for Applying, Let's Guide Innovation Forward! - Dear Aryan: Thank you for considering job id R173152, Data Science Intern with...	11/19/19
<input type="checkbox"/>	★	Bay Si (Nutanix, In.	Inbox	Nutanix, Inc. invitation for assessment - Hi Aryan Singh, Nutanix, Inc. has invited you to take an assessment. The details are below - - ...	11/18/19
<input type="checkbox"/>	★	no-reply	Inbox	Thank You from Postmates Next Steps - Hi Aryan, Thanks for applying to Postmates, your application has been received! To get start...	11/14/19
<input type="checkbox"/>	★	Shannon, me 2	Inbox	Vestmark, Inc. - Phone Interview Availability - Hey Shannon Thank you for the update. I would be able to talk on Tuesday after 11:30 a...	11/14/19

Strava Interview Availability External Inbox x



Caitlin Bernstein <cberstein@strava.com>

Hi Aryan,

Thanks for your interest in the Software Engineer Intern, Android position at Strava. We're excited to move forward with your application to the next stage.

The next step will be a 45 minute technical interview (Remote, via Zoom video call) with one of our Software Engineers. Be prepared to discuss your projects, experiences, and the technologies you are familiar with. You will also work through a (Strava-specific) coding exercise with your interviewer, but nothing too difficult on the first interview!

To give you an idea of our entire process, there are two more steps after the first technical screen:

1. A final technical interview, a little more challenging than the first one- 45 minutes, via Zoom, with a different Strava Software Engineer.
2. A recruiter call with one of our recruiters, 30 minutes to hear about your interest in Strava, what you're looking for in your internship and go over some logistics.

To help us schedule your first interview, please let us know when you're available by selecting the online calendar link below (some options this week and next).

If you have a deadline or timeline we need to be aware of, please let us know!

We look forward to hearing from you!

Best,
Caitlin Bernstein
Recruiter
Strava

[Enter your availability now >](#)

Mon, Oct 7, 2019, 12:11PM ★ x e i

Don't stop at 1 email

- Email 100s of recruiters
- Ask for their calls and numbers
- Schedule a call if possible
- Ask for multiple recruiters
- Acceptance rate is usually low



Aryan Singh <aryansingh@umass.edu>

to Caitlin <

Hey Caitlin,

Hope you're doing good.

I've filled out the time for Friday next week. Please let me know if that time doesn't work.

Mon, Oct 7, 2019, 9:12PM ★ x e i

Arrow Electronics - Internship

External

Inbox



F

Fatima Manjra <fatima.manjra@arrow.com>

to me ▾

Mon, Dec 31, 2018, 7:58 AM



Hi Aryan,

Thank you for your interest in being part of the Arrow team! I would like to schedule a quick 30 minute phone call with you to speak about the Internship program and ask you a few questions. Here is a link to my calendar, please schedule a time that will work best for you: <https://calendly.com/fatima-manjra>

I look forward to speaking with you soon!

Fatima Manjra

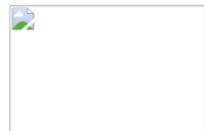
Recruiter

Arrow

Electronics, Inc.

fatima.manjra@arrow.com

www.arrow.com



Aryan Singh <aryansingh@umass.edu>

to Fatima ▾

Mon, Dec 31, 2018, 9:05 AM



I did it for friday, 4th at 2pm. Thanks a lot.

Hey Aryan,

We're excited to invite you to speak with members of our team! I'm working with Monica, and I've scheduled your phone interviews as follows:

 **1st Interview:** Thursday, January 9th, 2020 at 10:00am PT

Shared Doc: [Google Document One](#)

 **2nd Interview:** Thursday, January 9th, 2020 at 11:00am PT

Shared Doc: [Google Document Two](#)

 **Your Contact Number:** [+1 413-313-9365](tel:+14133139365)

On the day of your interview, please ensure that you are in a quiet area with reliable connection. Your interviewers will be calling this listed number at the specified time. Need to change your number? Notify us as soon as possible to update.

Before your interview:

- Let us know you can make it by replying-all to confirm. If you need to reschedule due to an unforeseen emergency, we're here to help. Please note that
- Complete our [Employment Application](#) (by today if you can!). This link should only be clicked once to generate your personal application. It is essential to your interviews and should not take more than 20 minutes. If you are unable to complete the form in one sitting, you can save your work by clicking the Finish button. Please carefully review your information to ensure everything is accurate and complete.
- It's important to us to create an accessible, inclusive workplace for everyone, so please complete [this form](#) if you need any accommodations for your interview, specialized equipment, or a lactation break and room. Our candidate accommodations team will then connect with you to confidentially discuss your options.

After your interview:

- Your recruiter will follow up with an update as soon as possible. If anything changes about your candidacy, please do the same for them.
- As a friendly reminder, our interview questions are confidential, so please keep things under wraps.

Good to know:

Test yourself with this email response

- How will you reply
- What will you say

Think about:

- What do we want next
- Simple and straightforward
- Good n bad examples

Hello Ben, hope you're doing well.

I've cleared all the technical rounds and am in the project search (host matching) part of the Google SWE intern process. I would love to work for your team ([Google Drive](#)) as an intern, if you're hiring.

My prior experience has been in mobile, web (full stack), backend, frontend, systems development where I've tried APIs, networking, databases, testing, UI/UX, and cloud services like firebase, storage, security, AWS etc. I'm also interested in learning new skills and technologies during my [internship](#).

You can find my open source projects at <https://github.com/aryansingh12>. I've attached my resume for your reference and look forward to your response. Thank you.

Regards,

Aryan Singh

Computer Science

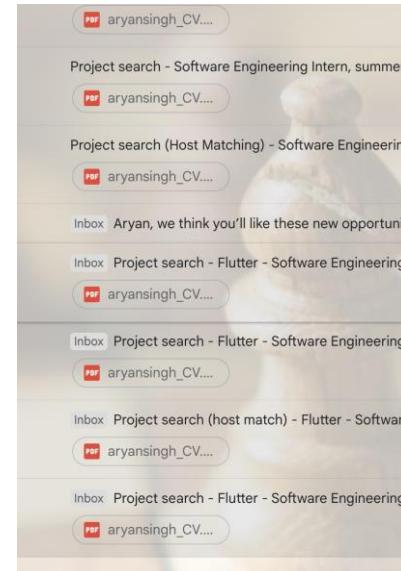
University of Massachusetts Amherst

[LinkedIn](#), [GitHub](#)

My email for TEAM MATCHING at Google

- After my selection
- To find a team
- Email 50+ people

So many people wait and get rejected in this round. This is how you become better than others.





Aryan Singh <aryansingh@umass.edu>

to mwilliamson ▾

Hey Marc,

Hope you're doing well. I'm a computer science student at the University of Massachusetts, Amherst, and am really interested in the SDE **internship** opportunity at Paypal. I applied some time back for the position and am waiting for the next steps.

This past summer, I was a Software Engineer Intern at Arrow Electronics, a large distributor of electronic components. I also work as a part-time mobile development engineer for a startup with over \$110k in funding. I've won various hackathons in the past, with projects like Safety for Women, Medical Tracker, and VR Classroom for Students with Disabilities.

“Applied sometime back”

- Revisit the emails
- Email them again
- Ask for feedback
- Ask for updates
- Ask for timelines

Facebook Software Engineering Intern : Aryan Singh



Aryan Singh <aryansingh@umass.edu>

to artip ▾

Thu, Dec 12, 2019, 6:00 AM



Hey Arti,

Hope you're doing well. I'm a computer science student at the University of Massachusetts, Amherst, and am really interested in the SDE **internship** opportunity at Facebook. I applied with a referral for the SDE intern position and am waiting for the next steps.

This past summer, I was a Software Engineer Intern at Arrow Electronics, a large distributor of electronic components. I also work as a part-time mobile development engineer for a startup with over \$110k in funding. I've won various hackathons in the past, with projects like Safety for Women, Medical Tracker, and VR Classroom for Students with Disabilities.

“I’ve applied with referral already”

- Tell this to recruiter
- Don’t wait and watch
- Alert them before others

You need very STRONG POINTS



Aryan Singh <aryansingh@umass.edu>
to Tenthbit+TNTH0774 ▾

Thu, Nov 14, 2019, 7:54 PM



Hey,

I'm an undergraduate at University of Massachusetts Amherst and am looking for an internship opportunity for summer 2020.

I'm really passionate about computer science and have developed a lot of Android mobile applications in the past. Being an Android developer, your app looks really impressive and I would love to add my contribution to it. I've done 2 internships previously, both in the space of Mobile development.

I've attached my resume and look forward to hearing back soon!

Regards,
Aryan Singh

One attachment • Scanned by Gmail ⓘ



Author:	Aryan Singh	(CV-101002)
Abstract: Aryan Singh is a highly experienced professional with a demonstrated history of working in the software development industry. Skilled in Java, Spring, Angular, and Python. Strong analytical and problem-solving skills. Proven ability to lead and manage teams effectively. Passionate about learning new technologies and staying updated with industry trends.		 Aryan Singh <small>Software Developer</small>
Background:	Software Development	
<ul style="list-style-type: none"> - Working as a Software Engineer at TechCorp Solutions Pvt. Ltd., developing web-based applications using Java and Spring Boot. - Applying Agile methodologies to manage projects and ensure timely delivery. 		
Major Projects:	Project Alpha - 2018 - present	
<ul style="list-style-type: none"> - Developed a complex web application for managing customer data and sales reports. - Utilized Java, Spring Boot, and PostgreSQL to build a highly scalable system. - Integrated machine learning models for predictive analysis. - Ensured high performance and reliability through continuous monitoring and optimization. 		
Skills & Abilities:	<ul style="list-style-type: none"> - Proficient in Java, Spring, Angular, and Python. - Excellent communication skills, both written and verbal. - Strong analytical and problem-solving skills. - Experienced in leading and managing software development teams. - Continuous learner, always seeking opportunities for growth and improvement. 	
PDF:	aryansingh_CV.pdf	

Don't make it TOOOOO SIMPLE.

- This is v simple and boring.
 - No links, github, portfolio, etc. BAD EMAIL



Matthew Dinh <mdinh@atlassian.com>

to me ▾

Hey Aryan!

Thanks for reaching out, it says in our system that you passed your hackerrank, so you should be hearing from our Campus Recruitment Team very shortly :)

Cheers

Matty

...

Thu, Dec 12, 2019, 9:40AM



Aryan Singh <aryansingh@umass.edu>

to Matthew ▾

Hey Matthew,

Thanks for the update. I'm looking forward to it!

Regards,

Aryan Singh

Computer Science

University of Massachusetts Amherst

[LinkedIn](#), [GitHub](#)

...

Thu, Dec 12, 2019, 11:20AM



Aryan Singh <aryansingh@umass.edu>

to Matthew ▾

Hey Matthew

Hope you're doing well. I'm still waiting to hear back from Atlassian for the next steps. I have an offer from a tier-1 company but I'm really interested in working with Atlassian and hope that something works out as soon as possible.

Fri, Dec 27, 2019, 5:00AM



REGULAR FEEDBACK + TIMELINE

C

Cristina Harris <cristina.harris@sonos.com>

to me ▾

Hello Aryan,

I hope all is well! My name is Cristina and I work at Sonos. A couple months back you applied to a software dev [internship](#) at Sonos. Are you still interested in it?

If so, please provide your availability and I can get you set up to speak with someone on the team!

Chat soon,
Cristina

A

Aryan Singh <aryansingh@umass.edu>

to Cristina ▾

Hey Cristina,

Hope you're doing well. I really appreciate you giving me the chance to schedule the interview but I've accepted an [internship](#) offer recently.

I would be willing to apply again for the new grad position starting next summer. Thank you.

Regards,
Aryan Singh
Computer Science
University of Massachusetts Amherst
[LinkedIn](#), [GitHub](#)

NEVER SAY NO

- I said no to internship but still willing to get full time later
- Never just say no
- I had the google offer so said no

Always

- Interview (never know about layoffs)
- Ask for personal contact
- Keep good relations
- Never be rude
- Never say no
- Always reply even if you don't 'have' to

Resources

- <https://www.30dayscoding.com/blogs/cold-email>
- Chat GPT
- Try and test
- Experiment
- Be creative
- Be unique

Find YOUR email and find your own journey

Don't try to find the “BEST” email for you

Welcome from the Google Intern Team! Inbox x



Hi Aryan,

We're looking forward to welcoming you for your **internship** soon. As you prepare for your first day, we wanted to share some resources, tips, and reminders to set you up for success.

Please reach out to us if you have any questions, we'll see you in a few days!

- The Intern Team

Before your first day

- **Getting Started:** You will be receiving an email from candidate-help@google.com with information on onboarding logistics, including how to set up your permanent Google account and the equipment you'll receive (please keep the box your equipment arrives in to use for return at the end of

You will see this one day VERY
VERY SOON!

**Join the community:
30DaysCoding.com for \$5/month**

- Free 1:1 with me
- Free resources every week
- Free events every week
- Amazing people
- Exclusive videos and courses

Coding projects + how to build awesome products

Frontend, full stack, Blockchain, AI, etc

1. Work on your ideas
2. Do not watch a tutorial unless NEEDED
3. Find blogs and guides before videos
4. Find interesting projects
5. Find new ideas
6. Work on trending things
7. Try to make money with projects
8. Try to turn projects into products
9. Go to hackathons and build something
10. Try to win in hackathons
11. Collaborate with people
12. Stop building alone all the time

Projects + products I've built

- [GetStartupNews.com](#) : AI-Powered Startup News for Social Media
- [KontentCreator.com](#) : Turn Your Blogs & Essays into Engaging Social Content
- [30DaysCoding.com](#) : Curate Coding Resources with Your Knowledge
- [BlockTrain.info](#) : Web3 Education Website
- [SEOPlatter.com](#) : Become an SEO Hero with AI
- [DSArevision.com](#) : Revision and other education guides
- [BabySaas.com](#) : Agency landing page website

And many more. 20+ domains

5 more projects - explore and see the quality

- <https://project-erc20.vercel.app/>
- <http://openriver-thirdweb.vercel.app/>
- <https://dopp.vercel.app/>
- <https://dev-pool.vercel.app/>
- <https://guruji.vercel.app/>

Learn REACT **BEFORE** anything else (if you're confused)

1. Very high in demand
2. Lot of packages and libraries
3. Very easy to make simple projects
4. Lot of flexibility
5. Next JS is built on top of it
6. You become independent developer
7. You can work on your ideas
8. Blockchain, AI, etc - everything needs a frontend

Full stack roadmap

- How to become a full stack king
- React / Next JS
- Firebase Supabase
- Auth, DB, API
- Depends on type of app
 - AI / Blockchain apps
- Mongo DB / Node / Express
- Make projects
 - product hunt, YC
- Sending emails
- Stripe / payment
- Ruby on rails
- Next JS - frontend and backend both
- Upstash, redis
- Flask / Django
- Deployment
- Users

Why do we need projects????

#1

- Become independent
- Implement ideas
- Enjoy coding
- Ship products
- Make money

#2

- Resume
- Interviews
- Startups

#3

- Essence of coding
- You learn a lot more

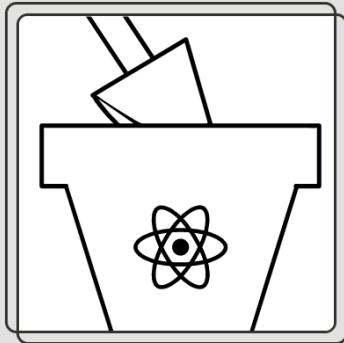
React app thinking process - learning

- learn basics / language
- Routing n navigation
- Jsx and data
- Props and state
- Deploy app
- Hooks
- Api / backend simple
- Make projects now
- Storage - firebase n supabase
- Deploy more apps
- Other libraries

Making a full stack app

- ▶ Thinking of idea
- ▶ Wireframe or design
- ▶ Create next app
- ▶ Routing and navigation
- ▶ Simple design
- + ::▶ Functionality 1
- ▶ Authentication if needed
- ▶ Hooks, API, etc
- ▶ Simple API folder if needed
- ▶ State management
- ▶ Props and passing data
- ▶ Making helper components
- ▶ Deploy

THINK AND THEN GET TO WORK



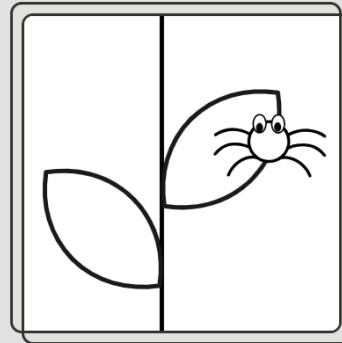
Part 0

Basics of web application operation



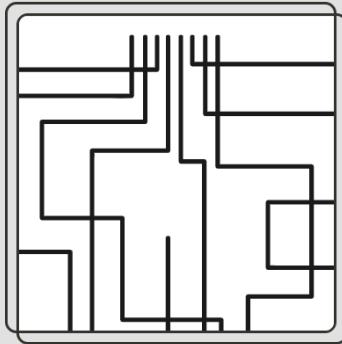
Part 1

Basics of React



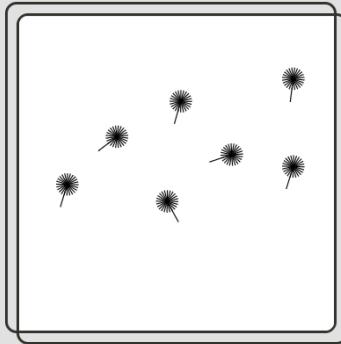
Part 2

Communication with the server



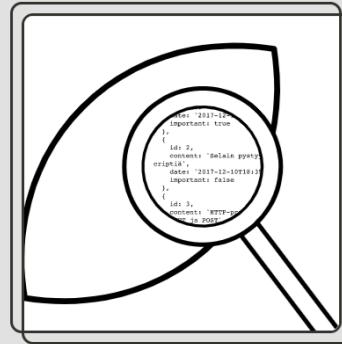
Part 3

Server programming with NodeJS
Express library



Part 4

Testing Express applications, user
management



Section 5

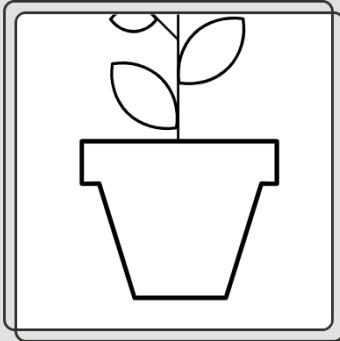
Testing a React application

Full stack open Part 1

- <https://fullstackopen.com/>

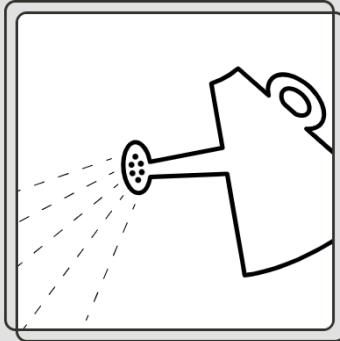
Complete 6 steps

- Basic web dev
- React
- Node, express
- Make apps based on this



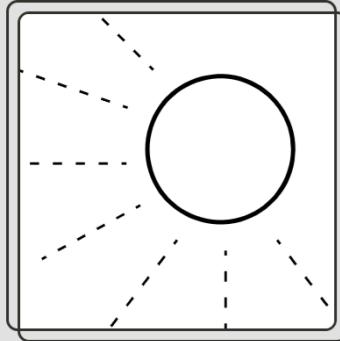
Section 6

Advanced space management



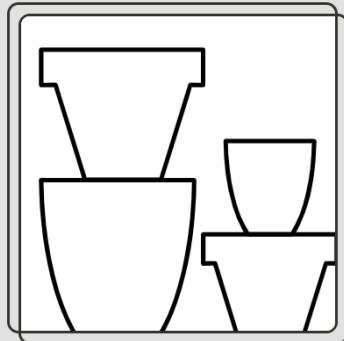
Section 7

React router, custom hooks, style libraries and webpack



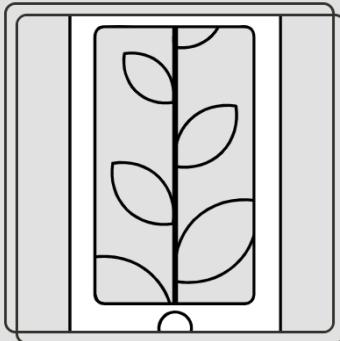
Section 8

GraphQL



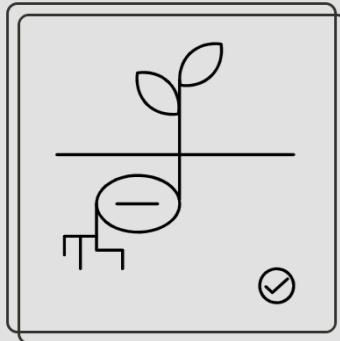
Section 9

TypeScript



Section 10

React Native



Section 11

CI/CD

Full stack open Part 2

- <https://fullstackopen.com/>

Complete next 6 steps

- Space
- Router
- GraphQL
- Typescript
- React Native
- CI CD

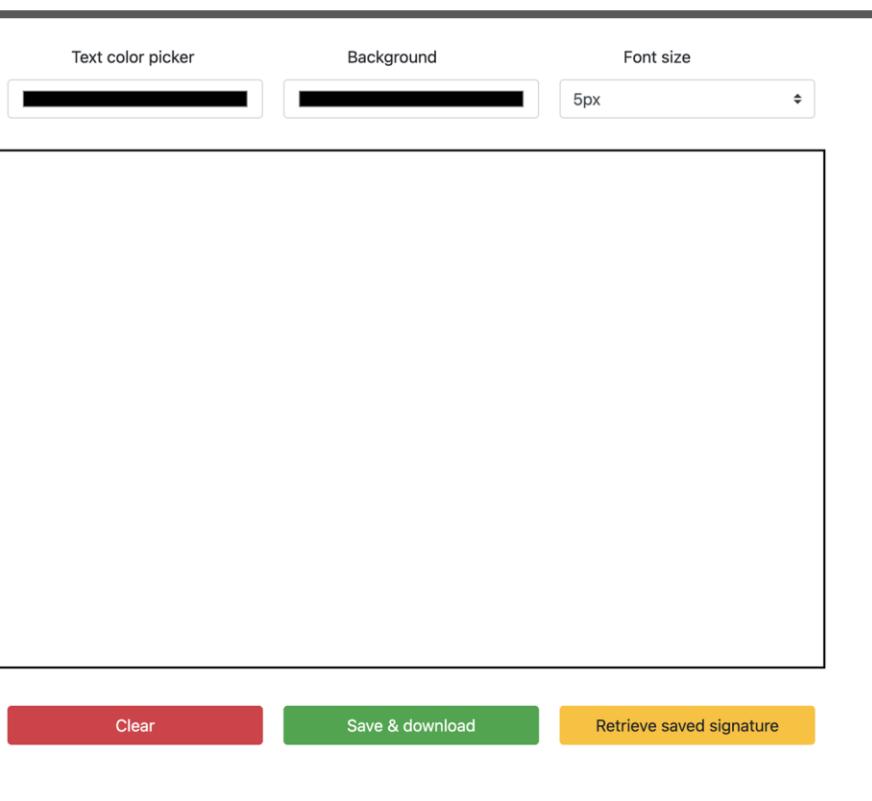
You can also use

- Firebase, supabase, appwrite
- Use that and deploy apps

STOP WATCHING ENDLESS TUTORIALS

Start building and shipping - let's END tutorial hell

Make projects with Chat GPT - <https://quicksign.me/> - learn prompt engineering



```
const colorPicker = document.getElementById('colorPicker');
const canvasColor = document.getElementById('canvasColor');
const canvas = document.getElementById('myCanvas');
const undoButton = document.getElementById('undoButton');
const clearButton = document.getElementById('clearButton');
const saveButton = document.getElementById('saveButton');
const fontPicker = document.getElementById('fontPicker');
const textInput = document.getElementById('textInput');
// const downloadButton = document.getElementById('downloadButton')
const fontSizePicker = document.getElementById('fontSizePicker');

// const lineWidthSlider = document.getElementById('lineWidthSlider');
const ctx = canvas.getContext('2d');

colorPicker.addEventListener('change', (event) => {
  ctx.fillStyle = event.target.value;
  ctx.strokeStyle = event.target.value;
});

canvasColor.addEventListener('change', (event) => {
  ctx.fillStyle = event.target.value;
  ctx.fillRect(0, 0, 800, 500);
});

canvas.addEventListener('mousedown', (event) => {
  isDrawing = true;
  lastX = event.offsetX;
  lastY = event.offsetY;
});

canvas.addEventListener('mousemove', (event) => {
  if (isDrawing) {
    ctx.beginPath();
    ctx.moveTo(lastX, lastY);
    ctx.lineTo(event.offsetX, event.offsetY);
    ctx.stroke();

    lastX = event.offsetX;
    lastY = event.offsetY;
  }
});

canvas.addEventListener('contextmenu', (event) => {
  event.preventDefault();
});
```



Filter

2D breakout game using Phaser

[Next »](#)

In this step-by-step tutorial, we create a simple mobile [MDN Breakout](#) game written in JavaScript, using the [Phaser](#) framework.

Every step has editable, live samples available to play with, so you can see what the intermediate stages should look like. You will learn the basics of using the Phaser framework to implement fundamental game mechanics like rendering and moving images, collision detection, control mechanisms, framework-specific helper functions, animations and tweens, and winning and losing states.

To get the most out of this series of articles you should already have basic to intermediate [JavaScript](#) knowledge. After working through this tutorial, you should be able to build your own simple Web games with Phaser.

Points: 20

Lives: 3

READ BLOGS AND GUIDES - stop copying and pasting code from youtube videos without thinking

- Reading is much better
- You will run into errors
- You should change according to your own ideas

- https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D_breakout_game_Phaser
- https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D_breakout_game_Phaser
- <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>
- <https://www.aha.io/blog/text-editor>

Find your Template

Jumpstart your app development process with our pre-built solutions.

The screenshot shows the Vercel Templates page with four template cards:

- Next.js Boilerplate**: A card for a Next.js application. It includes a preview image of a dark-themed dashboard, a brief description, and a "View on Vercel" button.
- Image Gallery Starter**: A card for an image gallery built on Next.js and Cloudinary. It includes a preview image showing a grid of photos, a brief description, and a "View on Vercel" button.
- Next.js AI Chatbot**: A card for a full-featured, hackable Next.js AI chatbot built by Vercel Labs. It includes a preview image of a conversational interface, a brief description, and a "View on Vercel" button.
- Commerce**: A card for a commerce application. It includes a preview image showing a product listing for a hoodie, a brief description, and a "View on Vercel" button.

Find a template

<https://vercel.com/templates>

- Implement ideas
- Ship fast
- Deploy fast
- Spread the word

AND THEN

- Add features
- Add auth
- Add other things

THIS WILL CREATE INTEREST

Generate the best marketing copy

Put an end to your creative block, get help from
your AI creative writer

Get your **free account today**

[Sign up with Github](#)

No credit card required

Build this full stack app from the template

- <https://github.com/tierrun/tier-vercel-openai/tree/main>
- Add your features
- Add something else
- Add a tadka taste

Some ideas

- Traveling, calories, fitness, jobs, etc with AI

**Join the community to find such awesome projects every week
with exclusive projects - 30dayscoding.com**

STEPS TO ELIMINATE TUTORIAL HELL AND SHIP FAST

1. Work on your ideas
2. Do not watch a tutorial unless NEEDED
3. Find blogs and guides before videos
4. Find interesting projects
5. Find new ideas
6. Work on trending things
7. Try to make money with projects
8. Try to turn projects into products
9. Go to hackathons and build something
10. Try to win in hackathons
11. Collaborate with people
12. Stop building alone all the time

- 17  **Project 16: No code app - internship list**
Aryan Singh • 7 views • 1 month ago
NO CODE TOOLS! WiX 11:54
- 18  **Project 17: Streamlit AI app**
Aryan Singh • 5 views • 1 month ago
Streamlit 3:33
- 19  **Project 18: Vercel, Next JS templates**
Aryan Singh • 4 views • 1 month ago
NEXT 4:31
- 20  **Project 19: Indie hackers overview**
Aryan Singh • 2 views • 4 weeks ago
INDIE HACKERS 4:49
- 21  **Project 20: Twitter for developers**
Aryan Singh • 8 views • 4 weeks ago
Twitter 5:02
- 22  **Project 21: Coding roadmaps**
Aryan Singh • 27 views • 3 weeks ago
Coding Roadmaps 12:29
- 23  **Project 22: Deploying your apps**
Aryan Singh • 17 views • 3 weeks ago
Vercel 2:01
- 24  **Project 23 Figma landing pages**
Aryan Singh • 3 views • 3 weeks ago

- 5  **Project 3: Get startup news - RSS / AI app**
Aryan Singh • 20 views • 1 month ago
RSS - AI APP
- 6  **Project 4: E-learning app - Next js, Firebase**
Aryan Singh • 30 views • 1 month ago
NEWS APP
- 7  **Project 5: Web3 full stack apps**
Aryan Singh • 26 views • 1 month ago
WEB3 TOOLS 8:59
- 8  **Project 6: Full stack developer course - fullstackopen.com**
Aryan Singh • 63 views • 1 month ago
React 5:16
- 9  **Project 7: Next JS todo app**
Aryan Singh • 40 views • 4 months ago
NEXT JS TODO APP 5:15
- 10  **Project 8: APIs for full stack development**
Aryan Singh • 10 views • 1 month ago
BUILD W APIS 5:52
- 11  **Project 9: Youtube to content with AI**
Aryan Singh • 5 views • 1 month ago
YOUTUBE AI 5:02

Finding ideas and existing projects

Frontend, full stack, Blockchain, AI, etc

Learn, ship, deploy!



Build Reddit 2.0 clone with NextJS

❖ React, SQL, Supabase, Nex...

Start building



Build an Uber Eats Clone with React Native

❖ React Native, Javascript,...

Start building



Build an Uber Clone with React Native

❖ React Native, Javascript,...

Start building



Build Super Mario Bros, Zelda, and Space Invaders with Kaboom.js

[LoveCupid Kaboom.js](#)



Build a YouTube Clone with React

❖ Express, Node, JavaScript...



Build a Chat App with Stories Using the Flutter SDK

❖ Flutter, Dart...

Unlimited projects and ideas

- 30 days coding website
- Chat gpt
- Your problems

It's a great way to learn.

- 3D Renderer
- Augmented Reality
- BitTorrent Client
- Blockchain / Cryptocurrency
- Bot
- Command-Line Tool
- Database
- Docker
- Emulator / Virtual Machine
- Front-end Framework / Library
- Game
- Git
- Network Stack
- Neural Network
- Operating System
- Physics Engine
- Programming Language
- Regex Engine
- Search Engine
- Shell
- Template Engine
- Text Editor
- Visual Recognition System
- Voxel Engine
- Web Browser
- Web Server
- Uncategorized

Build complicated projects

- Shells
- Engines
- Games
- Git
- Your own language
- Your own OS

And much more

<https://github.com/codecrafters-io/build-your-own-x#build-your-own-shell>

- ▶ Content app - kontentcreator.com
- ▶ Linkedin generator
- ▶ Cv generator
- ▶ Resume maker
- ▶ youtube video maker
- ▶ Blockchain education - Blocktrain
- ▶ Education app - blocktrain
- ▶ Blog app - coder aryan
- ▶ Ai app - Streamlit
- ▶ Streamlit dashboard app
- ▶ Streamlit data charts apps
- ▶ Linkedin photoshot - leap
- ▶ Wallpaper app - leap
- ▶ news ai app
- ▶ Travel ai app
- ▶ Calories ai app
- ▶ Code together app
- ▶ No code apps
- ▶ Create tweets with ai app
- ▶ Create linkedin post with ai app
- ▶ Create ads with ai - google n fb

Community projects - 30dayscoding.com

- New tech stack
- Detailed blogs
- Detailed guides
- Video resources
- Idea to deployment

{30}

WHY TO JOIN?

- REACT and NEXT JS
- AI, BLOCKCHAIN, WEB3
- OPEN SOURCE PROJECTS
- FULL STACK PRODUCTS

What's New?

- 1:1 MEETINGS AND EVENTS
- 24/7 COMMUNITY
- WHATSAPP + DISCORD
- WEEKLY LIVE SESSIONS

SWE @ GOOGLE

Product hunt and YC - Startups and products

Frontend, full stack, Blockchain, AI, etc

Is the next 🧔 here?



BrewNote

AI-powered notes for user interviews

Free Design Tools

Featured ▾

▲
405



Minutes

Make your meetings async with voice and video

Free Options Productivity

▲
306



Insanely Cool Tools

Discover insanely cool tools for startup founders

Free Productivity

▲
269



ReadMe

Transform API docs to real-time, interactive developer hubs

Promoted

▲
1,220

YOUR PRODUCTS →

Daily Startup News
[Invite users to review](#)

DSA revision
[Invite users to review](#)

TOP REVIEWED PRODUCTS



Acorn Box
Acorn Protocol Mobile Wallet

★★★★★ 5/5



squabbles
A new kind of social media - A...

★★★★★ 5/5



Moonrig.io
A free easy-to-use crypto portfolio...

★★★★★ 5/5



Daily Startup News

Posted on May 8th, 2023

Last activity 12d ago



Blocktrain

Posted on December 10th, 2022

Last activity 3mo ago



Creators Day

Posted on February 14th, 2022

Last activity 9mo ago



Quick sign

Posted on December 9th, 2022

Last activity 5mo ago

Product hunt - game changer.

- Learn from products
- Get new ideas
- Post your ideas
- Help and collaborate

- Top Companies (Valuation) 341
- Top Companies (Revenue) 50
- Is Hiring 952
- Nonprofit 43

- Black-founded 240
- Hispanic & Latino-founded 410
- Women-founded 612

Batch

- All batches 4275
- S23 85
- W23 269
- S22 234
- W22 402
- S21 393
- W21 336
- S20 208

[See all options](#)

Industry

- All industries 4275
- ▶ B2B Software and Services 1901
- Education 184

Search...

Showing 40 of 1,000+ companies



Airbnb San Francisco, CA, USA

Book accommodations around the world.

Y W09 MARKETPLACE TRAVEL



DoorDash San Francisco, CA, USA

Restaurant delivery.

Y S13 MARKETPLACE E-COMMERCE



Coinbase Remote; San Francisco, CA, USA

Buy, sell, and manage cryptocurrencies.

Y S12 CRYPTO / WEB3

Why YC? We give our companies a disproportionate advantage. [Learn More](#)



Dropbox San Francisco, CA, USA

Backup and share files in the cloud.

Batch

- All batches 4275

- S23 85
- W23 269
- S22 234
- W22 402
- S21 393
- W21 336
- S20 208

[See all options](#)

Industry

- All industries 4275
- ▶ B2B Software and Services 1901
- Education 184
- ▶ Consumer 755
- ▶ Financial Technology 529
- ▶ Healthcare 507
- ▶ Real Estate and Construction 128
- ▶ Industrials 226

[See all options](#)

Y Combinator companies - lot of amazing startups

**Join our exclusive 24/7
community for \$5**

Frontend, full stack, Blockchain, AI, etc

- ▶ Content app - kontentcreator.com
- ▶ Linkedin generator
- ▶ Cv generator
- ▶ Resume maker
- ▶ youtube video maker
- ▶ Blockchain education - Blocktrain
- ▶ Education app - blocktrain
- ▶ Blog app - coder aryan
- ▶ Ai app - Streamlit
- ▶ Streamlit dashboard app
- ▶ Streamlit data charts apps
- ▶ Linkedin photoshot - leap
- ▶ Wallpaper app - leap
- ▶ news ai app
- ▶ Travel ai app
- ▶ Calories ai app
- ▶ Code together app
- ▶ No code apps
- ▶ Create tweets with ai app
- ▶ Create linkedin post with ai app
- ▶ Create ads with ai - google n fb

Community projects - 30dayscoding.com

- New tech stack
- Detailed blogs
- Detailed guides
- Video resources
- Idea to deployment

{30}

WHY TO JOIN?

- REACT and NEXT JS
- AI, BLOCKCHAIN, WEB3
- OPEN SOURCE PROJECTS
- FULL STACK PRODUCTS

What's New?

- 1:1 MEETINGS AND EVENTS
- 24/7 COMMUNITY
- WHATSAPP + DISCORD
- WEEKLY LIVE SESSIONS

SWE @ GOOGLE

Full stack projects / discussing



- ▶ Education cms
- ▶ Notion cms
- ▶ Low code tools
- ▶ Dashboards
- ▶ Job boards
- ▶ Admin tables
- ▶ Table backend
- ▶ Drag n drop tools - pika or shots
- ▶ Fintech app
- ▶ Trading app - upstock
- ▶ Text based apps - notion / editor
- ▶ Video streaming
- ▶ Video recording

Additional projects and ideas to build

- ▶ Game - shots
- ▶ Screenshot tool
- ▶ Discord bot
- ▶ Shopify app
- ▶ Slack bot
- ▶ Chrome extensions
- ▶ Twitter bots
- ▶ Insta bots
- ▶ Linkedin ke upar
- ▶ Twitter ke upar - banner bear
- ▶ Api projects - image apis do
- ▶ Backend - dub.sh style
- ▶ Delivery service bana skte
- ▶ Maps related projects
- ▶ Product hunt se nikaalo ideas
- ▶ Notion based apps
- ▶ Restaurant apps
- ▶ meditation apps
- ▶ Sports finding app - pickleball
- ▶ Gym partner app
- ▶ Coding live - leetcode

I am discussing all these projects in my community

Join if you're interested

No need if you can build projects outside

Come if you want to 10x your journey for \$5

- 17  **Project 16: No code app - internship list**
Aryan Singh • 7 views • 1 month ago
NO CODE TOOLS! WiX 11:54
- 18  **Project 17: Streamlit AI app**
Aryan Singh • 5 views • 1 month ago
Streamlit 3:33
- 19  **Project 18: Vercel, Next JS templates**
Aryan Singh • 4 views • 1 month ago
NEXT 4:31
- 20  **Project 19: Indie hackers overview**
Aryan Singh • 2 views • 4 weeks ago
INDIE HACKERS 4:49
- 21  **Project 20: Twitter for developers**
Aryan Singh • 8 views • 4 weeks ago
Twitter 5:02
- 22  **Project 21: Coding roadmaps**
Aryan Singh • 27 views • 3 weeks ago
Coding Roadmaps 12:29
- 23  **Project 22: Deploying your apps**
Aryan Singh • 17 views • 3 weeks ago
Vercel 2:01
- 24  **Project 23 Figma landing pages**
Aryan Singh • 3 views • 3 weeks ago

- 5  **Project 3: Get startup news - RSS / AI app**
Aryan Singh • 20 views • 1 month ago
RSS - AI APP
- 6  **Project 4: E-learning app - Next js, Firebase**
Aryan Singh • 30 views • 1 month ago
NEWS APP
- 7  **Project 5: Web3 full stack apps**
Aryan Singh • 26 views • 1 month ago
WEB3 TOOLS 8:59
- 8  **Project 6: Full stack developer course - fullstackopen.com**
Aryan Singh • 63 views • 1 month ago
React 5:16
- 9  **Project 7: Next JS todo app**
Aryan Singh • 40 views • 4 months ago
NEXT JS TODO APP 5:15
- 10  **Project 8: APIs for full stack development**
Aryan Singh • 10 views • 1 month ago
BUILD W APIS 5:52
- 11  **Project 9: Youtube to content with AI**
Aryan Singh • 5 views • 1 month ago
YOUTUBE AI 5:02

DSA masterclass

Become a DSA expert + a good problem solver. Tips from experts

Important points

1. DSA is about problem solving and not just leetcode
2. Use DSA as your advantage instead of complaining
3. No need for CP if you are not confident with 200-250 questions
4. No need of a course if you know the basics
5. Stop pasting solutions and watching videos without TRYING
6. Stop comparing your journey to others
7. No, you don't need to solve 1000 questions
8. No, you don't have to be a born genius

I cleared 40+ DSA interviews and you can too!

Table of Contents

The document is divided as follows.

Table of Contents.....	2
Complete Data structures and Algorithms Roadmap.....	3
Our Aim.....	3
Practice.....	3
Arrays.....	4
Introduction.....	4
Hash maps, tables.....	4
2 Pointers.....	5
Linked List.....	9
Sliding Window.....	13
Binary Search.....	18
Recursion.....	25
Backtracking.....	32
BFS, DFS.....	40
Dynamic Programming.....	52
Trees.....	63
Graphs.....	70
Topological Sorting.....	81
Greedy Algorithms.....	85
Priority Queue.....	88
Tries.....	93
Additional Topics.....	96
Kadane's algorithm.....	96
Dijkstra's algorithm.....	97
AVL Trees.....	98
Sorting.....	99
More.....	99
Additional Awesomeness.....	99

Linked List

Introduction

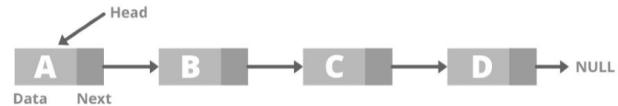
* Linked list is a data structure which stores objects in nodes in a snake-like structure. Instead of an array (where we have a simple list to store something) we have nodes in the linked list. It's the same thing though, you can store whatever you want - objects, numbers, strings, etc.

The only difference is in the way it's represented. It's like a snake: with a head and tail, and you can only access one thing at a time - giving its own advantages and disadvantages. So if you want to access the 5th thing, you can't do `linked_list[5]`, instead -> you would have to iterate over the list from the beginning and then stop when the number hits 5.

9

BlockTrain.info

Singly Linked List



Problem 1: Linked list methods

Here's how a linked list looks like: [Linked list: Methods](#)

Arrays

Introduction

* Informally, an array is a list of things. It doesn't matter what the things are; they can be numbers, words, apple trees, or other arrays. Each thing in an array is called an *item* or *element*. Usually, arrays are enclosed in brackets with each item separated by commas, like this: [1, 2, 3]. The elements of [1, 2, 3] are 1, 2, and 3.

- [Introduction to Arrays](#)
- <https://www.cs.cmu.edu/~15122/handouts/03-arrays.pdf>
- [An Overview of Arrays and Memory \(Data Structures & Algorithms #2\)](#)
- [What is an Array? - Processing Tutorial](#)

Arrays are used with all different types of data structures to solve different problems, so it's kind of hard to come up with array questions with just an array logic. Let's discuss some of the most famous patterns which concern arrays most of the time.

2D matrices are also arrays and are very commonly asked in interviews. A lot of graph, DP, and search based questions involve the use of a 2D matrix and it's important to understand the core concepts there. We've discussed the common patterns in each section below so make sure to check that out.

Hash maps, tables

* A hash table is a data structure that implements an associative array abstract data type, a structure that can map keys to values. In other words, we can store anything in the form of key value pairs.

Example: `map<string, string>`, means that this is a hashmap where we store string key and value pairs.

Problem 4: Merge sorted lists

Merge Two Sorted Lists

* We have 2 sorted lists and we want to merge them into one.

Does sorting tell you something? The element at the head would be the smallest.

Can we compare the heads every time and add those to the new list?

Ooooo, maybe yeah. Let's try comparing and then move the counter of the bigger element.

```
if l1.val < l2.val:  
    cur.next = l1  
    l1 = l1.next
```

Otherwise, we do this for the other node (because that's smaller)

```
else:  
    cur.next = l2  
    l2 = l2.next
```

What do we do once a list is done and the other one is left? Simply move the new linked list to the next pointer -> `cur = cur.next` and then add all the elements of the left over list.

```
cur.next = l1 or l2 # iterate over and add all the elements  
return head (the temporary new list that we made)
```

Problem 5: Merge K Sorted lists:

Leetcode 23. Merge k Sorted Lists

* We have k lists and we want to merge all of those into a big one.

1. One simple way would be to compare every 2 lists, call this function, and keep doing until we have a bigger list. Any other shorter way?
2. We can be smart about it and add all the lists into one big array or list -> sort the array -> then put the elements back in a new linked list!

Read

- [Linked list: Methods](#)
- [How I Taught Myself Linked Lists. Breaking down the definition of linked list](#)
- [Introduction to Linked List](#)

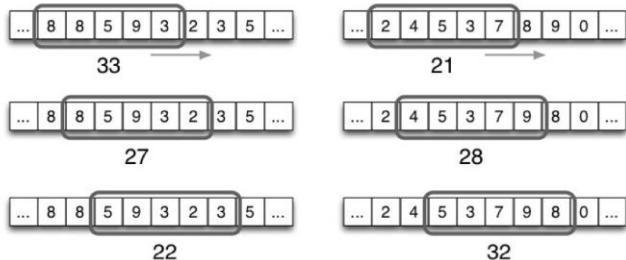
Videos

- [Data Structures: Linked Lists](#)
- [Interview Question: Nth-to-last Linked List Element](#)

Questions

- [141. Linked List Cycle \(Leetcode\)](#)
- [Delete Node in a Linked List](#)
- [19. Remove Nth Node From End of List](#)
- [Merge Two Sorted Lists](#)
- [Palindrome Linked List](#)
- [141. Linked List Cycle \(Leetcode\)](#)
- [Intersection of Two Linked Lists](#)
- [Remove Linked List Elements](#)
- [Middle of the Linked List](#)
- [lc 23. Merge k Sorted Lists](#)
-

the string/array.



Sliding window is a 2 pointer problem where the front pointer explores the array and the back pointer closes in on the window. Here's an awesome visualization to understand it more:

[Dynamic Programming - Sliding Window](#)

Problem 1: Max sum for consecutive k

* We have an array [1,2,3,2,4] and k=2, we want to return the max sum of the array with size 2. Looking at this for the first time, I would think of a brute force way to calculate all the subarrays, find their sum, store the maximum, and return it. However, that's very expensive. We don't really need to explore all the subarrays. Or, we can do that in an easier way (which is also cheaper): SLIDING WINDOW.

This is how sliding window would work here:

- We start with a window of 'k' from the left.
- We plan to move it to the right until the very end
- We remove the leftmost element (from the window) and add the right one as we move to the left
- We store the sum for every window and then return the max at the very end.

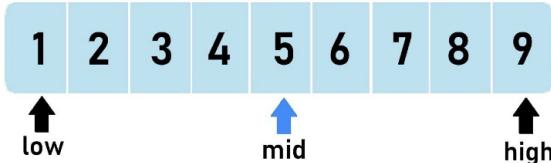
Storing the sum

- You can either calculate the sum every time -> which will be expensive
- Or we can just find the sum of the window the first time

Binary Search

Introduction

* We use binary search to optimize our search time complexity when the array is sorted (min, max) and has a definite space. It has some really useful implementations, with some of the top companies still asking questions from this domain.



The concept is: if the array is sorted, then finding an element shouldn't require us to iterate over every element where the cost is $O(N)$. We can skip some elements and find the element in $O(\log n)$ time.

Algorithm

* We start with 2 pointers by keeping a low and high -> finding the mid and then comparing that with the number we want to find. If the target number is bigger, we move right -> as we know the array is **sorted**. If it's smaller, we move left because it can't be on the right side, where all the numbers are bigger than the mid value.

Here's an iterative way to write the Binary search algorithm:

```
int left = 0, right = A.length - 1;
// loop till the search space is exhausted
while (left <= right) {
}
```

Problem 1: Max font size (Google internship)

* Google likes to test you on word problems with core principles. So even if they ask you a binary search question, it will be framed like a real life thing so that it's much harder to understand. They also test OOPS sometimes, by asking you to create classes and functions to display different things. Here's the question:

Given:

1. Height and width of a screen where you have to type
2. Height and width of each character you type on the screen
3. Min and max range of the font size of each character

Find the maximum font size such that the characters fit inside the screen

Once you understand the question, it's trivial to think of a brute force problem: explore all the possible font sizes and then see what fits at the end. Return that. Thinking a little more, we see that we have a range (sorted) and we don't really have to check for each font before choosing the maximum one. Shoot -> it's binary search.

Here's how the pseudo code looks like:

```
def find_max_font():
    max_font = 0
    start, end = min_font, max_font
    while start<=end:
        mid_font = start + (end-start)//2
```

BINARY SEARCH

RECURSION
it recurs.

it recurs.

RECURSION

"

* These are some questions I have when I look at a recursive question/solution, you probably have the same. Let's try to figure out them

- What happens when the function is called in the **middle** of the whole recursive function?
- What happens to the stuff **below** it?
- What do we think of the base case?
- How do we figure out when to **return** ?
- How do we save the value, specially in the **true/false** questions?
- How does **backtracking** come into place, wrt recursion?

Let's try to answer these one by one. A recursive function means that we're breaking the problem down into a smaller one. So if we're saying $\text{function}(x/2)$ -> we're basically calling the function again with the same parameters.

So if there's something below the recursive function -> that works with the same parameter. For instance, calling $\text{function}(x/2)$ with $x=10$ and then printing (x) after that would print 10 and then 5 and so on. Think of it as going back to the top of the function, but with different parameters.

The return statements are tricky with recursive functions. You can study about those things, but practice will help you get over it. For instance, you have fibonacci, where we want to return the sum of the last 2 elements for the current element -> the code is something like $\text{fib}(n) + \text{fib}(n-1)$ where $\text{fib}()$ is the recursive function. So this is solving the smaller problem until when? -> Until the base case. And the base case will return 1 -> because eventually we want the $\text{fib}(n)$ to return a number. This is a basic example, but it helps you gain some insights on the recursive part of it.

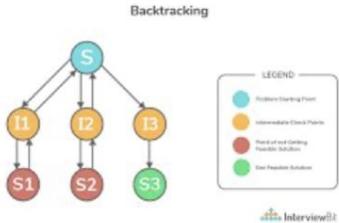
Something complex like dfs or something doesn't really return anything but transforms the 2d matrix or the graph.

RECURSION

- V v v important topic
- Read every line
- Understand every pattern

This is the basis for backtracking and DP - 2 very important topics

- Don't skip questions
- Make notes
- Solve 25+ problems



Problem 1: Generate parentheses

[22. Generate Parentheses](#)

★ Generate balanced parentheses, given a number.

In simple words, we want to print out all the possible cases -> valid parentheses can be generated.

One thing which strikes me is -> we need a way to add "(" and ")" to all possible cases and then find a way to validate so that we don't generate the unnecessary ones.

The first condition is if there are more than 0 open / left brackets, we recurse with the right ones. And if we have more than 0 right brackets, we recurse with the left ones. Left and right are **initialized** at N - the number given.

READ AND UNDERSTAND PATTERNS.

- Copy pasting doesn't help
- Reading doesn't help
- Understanding helps

Problem 3: Letter combination of phone numbers

[17. Letter Combinations of a Phone Number](#)

* Interesting problem and can be solved both iteratively and recursively (same for any problem). The first thing which comes to mind is to have a map of the numbers and digits, so that we actually use it. The second thing which is trivial is that -> we would iterate over, take all the possible ways, and then store it in a list. It's basically a cliche backtracking problem where we have some arrays and we want all the possible cases in those.

A recursive function would need to have something in the arguments which we add + we the array (using python sub-array)

```
combo(combination+letter, digits[1:])
```

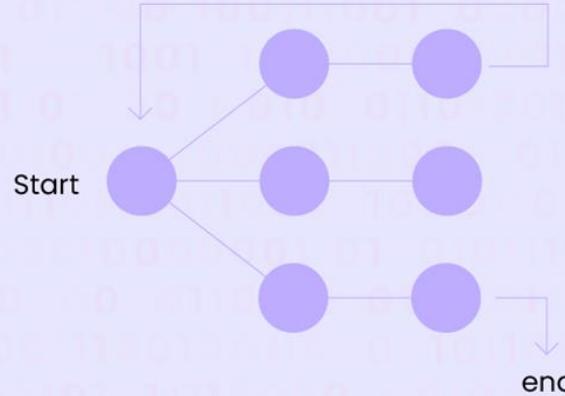
We do this for every letter and add a base case for adding the combination to the result a Here's how the complete code looks like

```
def combo(combination, digits):
    if len(digits)==0:
        a.append(combination)

    else:
        for letter in phone[digits[0]]:
            combo(combination+letter, digits[1:])
```

Here's a java solution code for it: [My recursive solution using Java](#)

BACKTRACKING



Introduction

* Backtracking can be seen as an optimized way to brute force. Brute force approaches evaluate every possibility. In backtracking you stop evaluating a possibility as soon as it breaks some constraint provided in the problem, take a step back and keep trying other possible cases, see if those lead to a valid solution.

Think of backtracking as exploring all options out there, for the solution. You visit a place, there's nothing after that, so you just come back and visit other places. Here's a nice way to think of any problem:

- Recognize the pattern
- Think of a human way to solve it
- Convert it into code.

Problem 1: Permutations

[46. Permutations](#)

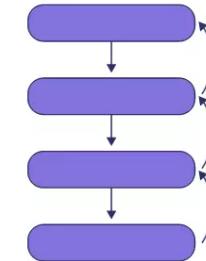
* We have an array [1,2,3] and we want to print all the possible permutations of this array. The initial reaction to this is - explore all possible ways -> somehow write 2,1,3, 3,1,2 and other permutations.

Second step, we recognize that there's a pattern here. We can start from the left - add the first element, and then explore all the other things with the rest of the items. So we choose 1 -> then add 2,3 and 3,2 -> making it [1,2,3] and [1,3,2]. We follow the same pattern with others.

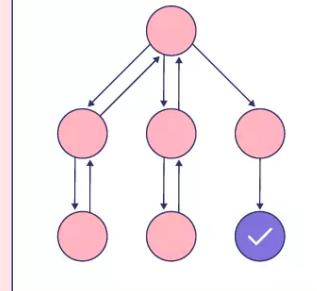
How do we convert this into code?

- Base case
- Create a temporary list
- Iterate over the original list
 - Add an item + mark them visited
 - Call the recursive function
 - Remove the item + mark them unvisited

Recursion



Backtracking



Problem 3: Combination Sum

39. Combination Sum

* We want to return the numbers which would add up to the target number given. We have to return all the possible combinations. So this is basically all subsets (with repeats allowed) with a target given.

From the get go, I know one thing -> we want to explore all cases, find the ones where the target matches, and then add that to a list, and return that list.

Backtracking template: Make a choice

- Iterate over the array
 - Add the item
 - Backtrack - recursive call
 - Remove the item

```
for(int i = start; i < nums.length; i++){  
    tempList.add(nums[i]);  
    backtrack(list, tempList, nums, target_left - nums[i], i); // not i + 1  
    because we can reuse same elements  
    tempList.remove(tempList.size() - 1);  
}
```

35

BlockTrain.info

A good thing to note here is that we pass in the `target_left - nums[i]` which basically means that we're choosing that element and then subtracting that from what we have in the argument. So the base case with this would be

`Target_left == 0` -> because that's when we know we can make the target.

One other thing to save some time and memory can be `target_left < 0` -> to return when we reach here, because negative numbers can never become positive numbers. So once the `target_left` is below 0, it can never come up -> good to just return;

```
public List<List<Integer>> combinationSum(int[] nums, int target) {
```

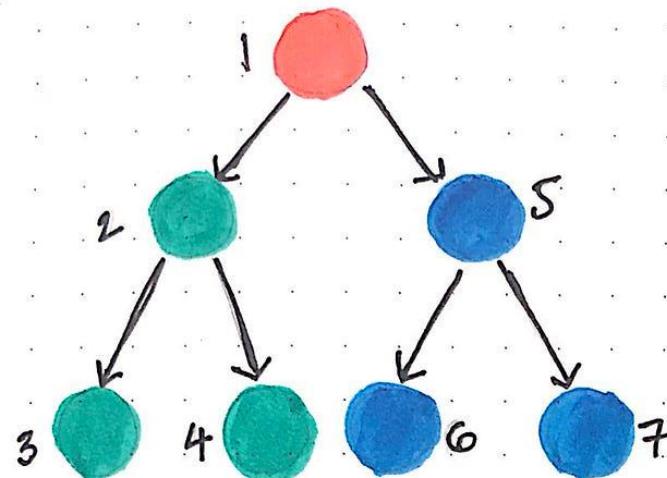
Backtracking is v v important pattern / algo / etc

- Used in real life use cases
- Finding cheapest flights/shoes/etc
- Finding fastest etc
- UNDERSTAND THE PATTERN HERE

Push, recurse, pop is a famous pattern

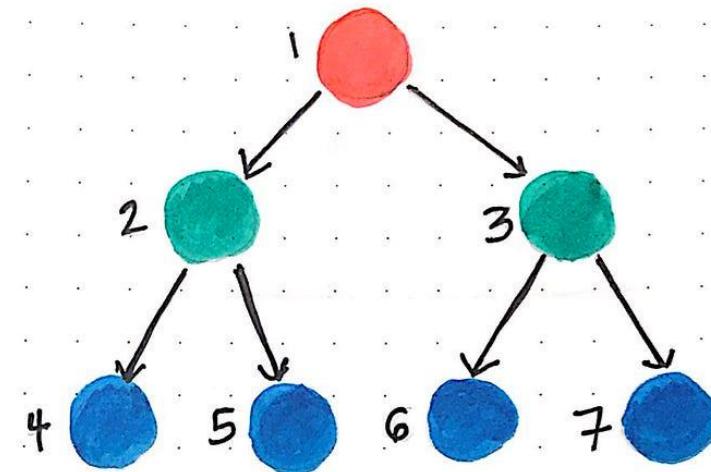
- We use the element
- Recurse and solve for it
- Then remove it

Read every word and line from this.



Depth-first search

- Traverse through left subtree(s) first, then traverse through the right subtree(s).



Breadth-first search

- Traverse through one level of children nodes, then traverse through the level of grandchildren nodes (and so on...).

Here's a beautiful visualization of a search in a tree: [Branch and Bound - Depth-Limited Search](#)

Here's a general iterative dfs pseudo-code template:

```
def dfs(root, target):
    stack = []
    stack.append(root) # add the first item

    while len(stack)>0:
        node = stack.pop() # pop the grid item

        if(node == target):
            return true

        # explore more
        # For trees -> if root.left or root.right
        if (condition):
            stack.append(new_item)

    return false;
```

The second step is that of MEMOIZATION and we want to keep a track of all the nodes visited when we're iterating over. Here's a complete version of a BFS algorithm where we keep track of the visited node using an array **discovered []**

This could be anything - array, map, set - depending on the situation. The only thing we need is to store the visited things so that we're not repeating any work.

```
public static void BFS(Graph graph, int v, boolean[] discovered)
{
    // create a queue for doing BFS
    Queue<Integer> q = new ArrayDeque<>();
```

Here are some implementations and use cases for DFS, BFS:

DFS:

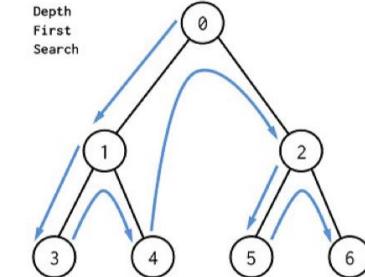
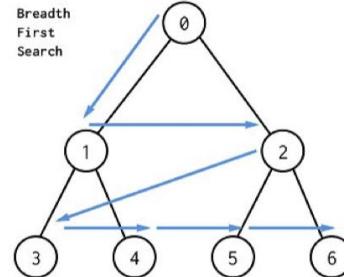
- Find connected components in a graph
- Calculate the vertex or edges in a graph
- Whether the graph is strongly connected or not
- Wherever you want to explore everything or maybe go in depth

BFS

- Shortest path algorithms and questions
- Ford fulkerson algorithm
- Finding nodes in a graph
- Wherever there is a shortest thing, finding something quickly, etc.

42

BlockTrain.info



```

public int numIslands(char[][] grid) {
    int count=0;
    for(int i=0;i<grid.length;i++){
        for(int j=0;j<grid[0].length;j++){
            if(grid[i][j] == '1'){
                dfs(grid, i, j);
                count+=1;
            }
        }
    }
    return count;
}

public void dfs(char[][] grid, int i, int j){
    if(i<0 || i>=grid.length || j<0 || j>=grid[0].length){
        return;
    }
    if(grid[i][j]== '1'){
        grid[i][j] = '#';
        dfs(grid, i+1,j);
        dfs(grid, i,j+1);
        dfs(grid, i,j-1);
        dfs(grid, i-1,j);
    }
}

```

The most important DFS problem - number of islands.

- Use a pen and paper
- Make a diagram
- Visualize how things are working

Solution

- Iterate, change the number
- Count the islands
- Change it back
- Do it for all the indexes

```

count=0

for i in range(len(grid)):
    for j in range(len(grid[0])):
        if grid[i][j]=='1':
            dfs(grid, i, j)
            count+=1

return count

def dfs(grid, i, j):
    s=[]
    s.append((i,j))
    while len(s)>0:
        a,b = s.pop()
        grid[a][b]='X'
        if a>0 and grid[a-1][b]=='1':
            s.append((a-1,b))
        if b>0 and grid[a][b-1]=='1':
            s.append((a,b-1))
        if a<len(grid)-1 and grid[a+1][b]=='1':
            s.append((a+1,b))
        if b<len(grid[0])-1 and grid[a][b+1]=='1':
            s.append((a,b+1))

```

```

public int numIslands(char[][] grid) {
    int count=0;
    for(int i=0;i<grid.length;i++){
        for(int j=0;j<grid[0].length;j++){
            if(grid[i][j] == '1'){
                dfs(grid, i, j);
                count+=1;
            }
        }
    }
    return count;
}

public void dfs(char[][] grid, int i, int j){
    if(i<0 || i>grid.length || j<0 || j>grid[0].length){
        return;
    }
    if(grid[i][j]== '1'){
        grid[i][j] = '#';
        dfs(grid, i+1,j);
        dfs(grid, i,j+1);
        dfs(grid, i,j-1);
        dfs(grid, i-1,j);
    }
}

```

ITERATIVE VS RECURSIVE

SOLVE USING BOTH

Problem 3: Rotten oranges

[994. Rotting Oranges](#)

* Every minute a fresh orange turns rotten if it's around a rotten orange. Similar to life -> if you're around negative people, you tend to be negative. Keep a positive outlook, help everyone, and take things forward!

This is an amazing question -> let's understand the iterative way of doing this and how to solve any searching related question with a stack or queue -> iteratively. We have the minimum condition here, so using BFS is the way to go! A simple pattern, as discussed before is:

- Prepare the stack/queue -> Add the initial nodes
- Pop the node from stack, mark it visited, add the valid neighbors
- Repeat the process for the new nodes.

First step is to prepare the queue. We add the rotten oranges (represented by 2) to the queue and also count the total number of oranges. 0 -> means an empty place.

```
for (int i = 0; i < grid.length; i++) {  
    for (int j = 0; j < grid[0].length; j++) {  
        if (grid[i][j] != 0) total++;  
        if (grid[i][j] == 2) q.offer(new Pair(i, j));  
    }  
}
```

We have the queue ready and now we iterate until it's empty: `while (stack.isEmpty()) {}`. We want to add all the neighbors of the current orange, which are in 4 directions and here's something to note when you have conditions like this.

When we want to traverse in all 4 directions, or maybe in 8 directions if we have a double condition, we can make a directions dictionary and iterate over it. Something like: [[0,1], [0,-1], [1,1], [1,0]] or `int[] dirs = {{1,0},{-1,0},{0,1},{0,-1}};`

```
while (! q.isEmpty()) {  
    int size = q.size();  
    rotten += size;  
    // if the total number of rotten oranges matches our local variable  
    // then return the time it took  
    if (rotten == total_rotten) return time;  
}
```

Read ↗

- [Leetcode patterns 1](#)

49

[BlockTrain.info](#)

- [Leetcode Patterns 2](#)

- [Depth-First Search \(DFS\) vs Breadth-First Search \(BFS\) – Techie Delight](#)

Videos 🎥

- [Breadth First Search Algorithm | Shortest Path | Graph Theory](#)
- [Depth First Search Algorithm | Graph Theory](#)
- [Breadth First Search grid shortest path | Graph Theory](#)

Questions 💬

- [Flood Fill](#)
- [Leetcode - Binary Tree Preorder Traversal](#)
- [Number of Islands](#)
- [Walls and Gates](#)
- [Max Area of Island](#)
- [Number of Provinces](#)
- [279. Perfect Squares](#)
- [Course Schedule](#)
- [C/C++ Program for Detect cycle in an undirected graph](#)
- [127. Word Ladder](#)
- [542. 01 Matrix](#)
- [Rotting Oranges](#)
- [279. Perfect Squares](#)
- [797. All Paths From Source to Target](#)
- [1254. Number of Closed Islands](#)

Dynamic Programming



Dynamic Programming

Introduction

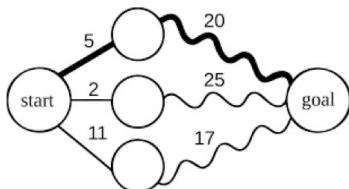
* Dynamic programming is nothing but recursion + memoization. If someone tells you anything outside of this, share this resource with them. The only way to get good at dynamic

50

BlockTrain.info

programming is to be good at recursion first. You definitely need to understand the magic of recursion and memoization before jumping to dynamic programming.

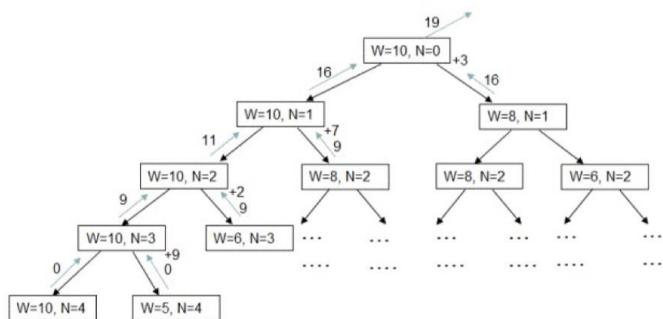
The day when you solve a new question alone, using the core concepts of dynamic programming -> you'll be much more confident after that.



So if you've skipped the recursion, backtracking, and memoization section -> go back and complete those first! If you've completed it, keep reading. You will only get better at dynamic programming (and problem solving in general) by solving more recursion (logical) problems.

Problem 1: 0-1 Knapsack

* This is the core definition of dynamic programming. Understanding this problem is super important, so pay good attention. Every problem in general, and all DP questions have a CHOICE at every step.



Recursion tree for 0-1 Knapsack problem

Thinking about the arguments, a good recursive function would be passing in the weights, values, index, and the remaining weight? That way `remaining_weight == 0` can be our base case. You can absolutely have other recursive functions with different arguments, it's about making things easier.

```
//include the ith item
int include = v[i] + knapsack(w, v, maxWeight - weights[i], i+1);
// don't include
int exclude = knapsack(weights, values, maxWeight, i+1);
```

We think of the base case now. A straightforward one looks like `maxWeight == 0`, which is also the REMAINING weight as we're subtracting the weight every time we're iterating with the included item.

The second one and the most usual one is when you reach the end of the array, so `index == weights.length`. Can also be `values.length` as they're the same.

Here's the code for it:

```
knapsack(weights [], values [], maxWeight = 0, index = i, memo_set = set())
{
    if(i == weights.length || maxWeight == 0){
```



DYNAMIC PROGRAMMING

DP Introduction

Dynamic Programming Playlist | Coding | Interview Questions |...

Aditya Verma

50 videos 8,160,685 views Last updated on Nov 1, 2020



Play all

Shuffle

This playlist explains Dynamic Programming in a concise way. Explaining how to approach a Dynamic Programming problem and moreover how to identify it first.

1 unavailable video is hidden

1	DYNAMIC PROGRAMMING DP Introduction 13:57	Dynamic Programming Introduction Aditya Verma • 1.2M views • 3 years ago
2	DYNAMIC PROGRAMMING Types of knapsack 13:50	2 Types of knapsack Aditya Verma • 377K views • 3 years ago
3	DYNAMIC PROGRAMMING 01 Knapsack Recur 21:04	3 01 Knapsack Recursive Aditya Verma • 477K views • 3 years ago
4	DYNAMIC PROGRAMMING 01Knapsack Memoiz 13:53	4 01Knapsack Memoization Aditya Verma • 389K views • 3 years ago
5	DYNAMIC PROGRAMMING Knapsack Tabula 41:08	5 01 Knapsack Top Down DP Aditya Verma • 485K views • 3 years ago
6	DYNAMIC PROGRAMMING Identification of Knapsack Problems 1 Introduction 6:17	6 Identification of Knapsack Problems and Introduction Aditya Verma • 219K views • 3 years ago
7	DYNAMIC PROGRAMMING Subset Sum Problem 13:57	7 Subset Sum Problem

- https://www.youtube.com/playlist?list=PL_z_8CaSLPWekghdCPmFohncHwz8TY2Go
- Best DP playlist
- For hindi - codebix

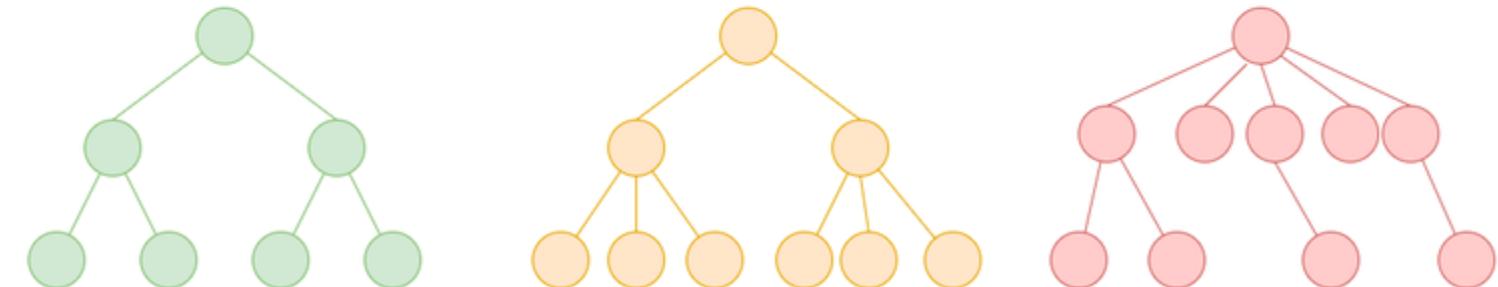
Trees

(on the basis of number of children)

Binary Tree

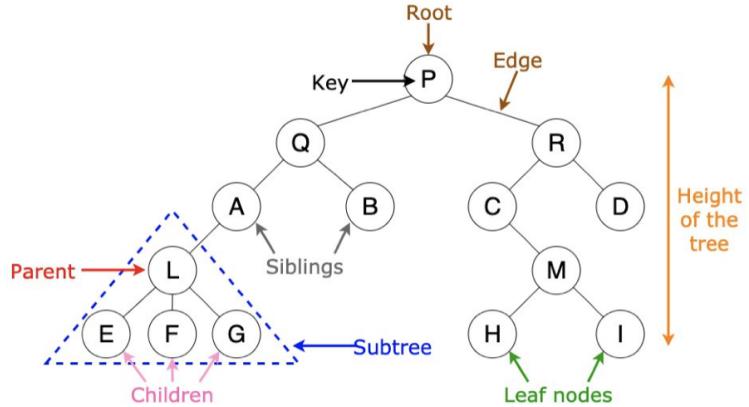
Ternary Tree

N-ary Tree



Introduction

* I love trees, but actual ones - not these. Just kidding, I love all data structures. Let's discuss trees. They're tree-like structures (wow) where we can store different things, for different reasons, and then use them to our advantage. Here's a nice depiction of how they actually look:



Recursion is a great way to solve a lot of tree problems, but the iterative ones actually bring out the beauty of them. Making a stack and queue, adding and popping things from that, exploring children, and repeating this would definitely make sure you understand it completely. You should be seeing this visually in your head, when you do it iteratively.

Pattern: Traversals

* There are 3 major ways to traverse a tree and some other weird ones: let's discuss them all. The most famous ones are pre, in, and post - order traversals. Remember, in traversals -> it's not the left or right node (but the subtree as a whole).

We start with the root, move until it's null or the stack is empty. We move to the left if we can, if not -> we pop, add the popped value and then move right.

```
List<Integer> res = new ArrayList<>();
if(root==null) return res;

Stack<TreeNode> stack = new Stack<>();
TreeNode curr = root;
while(curr!=null || !stack.isEmpty()){
    if(curr!=null){
        stack.push(curr);
        curr = curr.left;
    }else{
        curr = stack.pop();
        res.add(curr.val);
        curr = curr.right;
    }
}
return res;
```

Pre order traversal

* We add the root, then the left subtree, and then the right subtree. It's a stack so things work in the opposite direction -> first in last out, so make sure to check that carefully.

```
Stack<Node> stack = new Stack();
stack.push(root);
result = [];

while (!stack.empty())
{
    Node curr = stack.pop();
    result.push(curr.data);
    // print node
```

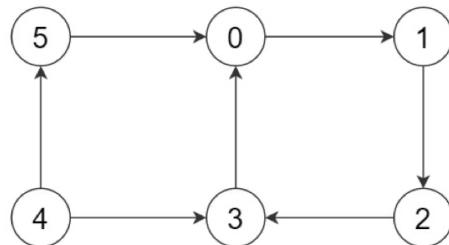
Graphs

Introduction

* A lot of graph problems are covered by DFS, BFS, topo sort in general -> but we're going to do a general overview of everything related to graphs. There are other algorithms like Djikstra's, MST, and others - which are covered in the greedy algorithms section.

A lot of graph problems are synced with other types = dynamic programming, trees, DFS, BFS, topo sort, and much more. You can think of those topics sort of coming under the umbrella of graph theory sometimes.

Problem 1: Finding the root vertex



* A human way of finding the root will be to look at 4 and say that there are no incoming edges at 4, so it's the root. Think of it in a tree like format, where the root is at the top and we have children below it.

How do we code this?

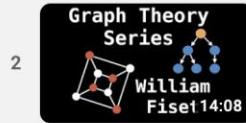
Graphs

- Searches
- Finding something
- Flights, comparison Qs
- Coloring, visited, etc
- Cyclic components
- Connected components



Graph Theory Algorithms

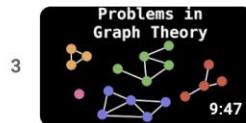
WilliamFiset • 173K views • 3 years ago



Graph Theory Series

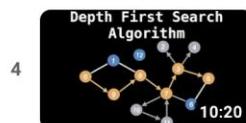
Graph Theory Introduction

WilliamFiset • 121K views • 5 years ago



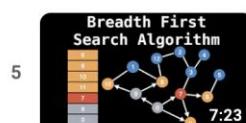
Overview of algorithms in Graph Theory

WilliamFiset • 72K views • 5 years ago



Depth First Search Algorithm | Graph Theory

WilliamFiset • 367K views • 5 years ago



Breadth First Search Algorithm | Shortest Path | Graph Theory

WilliamFiset • 532K views • 5 years ago



Breadth First Search grid shortest path | Graph Theory

WilliamFiset • 287K views • 5 years ago



Introduction to tree algorithms | Graph Theory

WilliamFiset • 65K views • 3 years ago

- <https://www.youtube.com/playlist?list=PLDV1Zeh2NRsDGO4--qE8yH72HFL1Km93P>
- William fiset - amazing graph

Topological Sorting	81
Greedy Algorithms	85
Priority Queue.....	88
Tries.....	93
Additional Topics	96
Kadane's algorithm.....	96
Djikstra's algorithm.....	97
AVL Trees.....	98
Sorting.....	99
More.....	99
Additional Awesomeness	99

ADDITIONAL TOPICS - IMPORTANT