[**https://stackoverflow.com/questions/62899578/making-sieve-of-eratosthenes-more-memory-efficient-in-python**](https://stackoverflow.com/questions/62899578/making-sieve-of-eratosthenes-more-memory-efficient-in-python)

**work on prime sieving, see if there is a quicker way to do segmented sieving than going through the list of previous primes.**

**Make web crawlers on ATP tour, build my own player watch list with ranking shown, build my own web crawler that’s always running in the background to check when a new player’s court level highlights are dropped, organize links of videos by player names, title of video, date, sorting by ascending order, if I get cracked, can write AI script to analyze the videos in the links, rate them based on algorithm to determine if video is shaky or not, high quality, whether I should watch or not. See if I can sell this to ATP Tour! Or make my own site, with subscription features.**

**Project Euler web scraper, parse the forum, build your own ranking by Kudos, activity, usertag mentions, and self-reported runtime (135 ms), take into account processor to see true algorithmic efficiency, place the top usertags in a list, search up usernames on search engine to see hit results, try to determine who the people are.**

**\*\*\*\*\*\*\* Great Idea From Mama! Make a highly flexible frontend where you can drag and drop items on the page, resize, and customize your page to how you want! And the next time you go to the webpage, it will save your customizations.**

**Could be very fun and wholesome to sit around as a family and have a good time whilst brainstorming software projects and thinking of interesting ideas, and discussing implementation of the ideas! Unique environment!**

**Cool idea: Make a visual, pdf-like, Microsoft Paint-like code text editor, where you can draw stuff, circle stuff, highlight stuff, not just commenting out with special comment characters. Can draw diagrams of how code works. When circling, heuristic code will identify what lines of code were circled, and deleting lines of code will result in the circle being updated, drawn to the screen, separate from the code file of course. The data will be saved in a graphics initialization file, monitor draws that to the screen, layered on top of the code. Should be a very challenging and creative project! In this age of burgeoning tech, should be many niche, simplistic IDEs/text editors to choose from! Could be popular amongst more artistic, graphic/visual learners, maybe cater towards girls? A great way to think about UI, UX, pleasing designs, interesting actions/algorithms. E.g. if you circle something in the integrated terminal and scroll enough, then circle should disappear, but then circle should reappear!**

**When you get cracked, try writing a language checker, checker basic grammar of subject/verb plurality, parallelism, tense agreement, like in a Word doc. Then for more advanced checkers, like Grammarly, pivot to natural language processing and artificial intelligence!**

**First write a machine-instruction-to-assembly converter. Then write an assembly to machine instructions converter. Then write a converter between ARM and MIPS assembler (an assembly-to-assembly assembler), and vice versa, and between x86 as a great programming exercise! Write them in C++ and Python.**

**Research on graph isomorphism, can an efficient algorithm be discovered?**

**Get into drone programming, drone AI, computer vision**

**Maybe teach AI music theory and feed it pieces of the greats, so great classical music can be made?**

**For polished software and project ideas, try to make outcome be as convenient as possible, reduce user mediation. Automate things as much as possible, but correctly! Should be convenient to use for the casual computer user. For instance, for software that calculates the Horton-Strahler number of a river (inspired by Donald Knuth webpage), we can let user type in the name of the river into our website. Or we can let user upload image of real river, map image, satellite image, vantage point 3D view (just a transformation (3D rotation of image) from bird’s eye to a more horizontal vantage point), and calculate Horton-Strahler number from there.**

**\*\*\* Multiple ideas for implementations of Horton-Strahler calculator software. Can try searching if number has already been calculated, if known streams/tributaries already exist, then do recursive search on the streams/tributaries (knowledge graph of online river search may show how many tributaries it has. i.e. river is a structure, vector of immediate tributes is an attribute, as is the total Horton-Strahler number). Can search for river on Google Maps, and then use computer vision to hone in on the river (and ignore other bodies of water) as a single connected component, then calculate Horton-Strahler number from there.**

**\*\*\* A lot of optimizations to reduce number of operations for a computer is to make use of predictive heuristics, make a bunch of guesses that are likely to be correct. If guess is not correct, then eliminated a possibility without spending too much time on it. So in essence, a hard-coded “sort” based on chance of occurring.**

**Tennis ball-tracking detection camera supporting bounce location, shot speed, and shot spin is a huge hit. Making AI tennis-playing robot would be an even bigger hit, with the correct marketing!**

**Exercises to do:**

**Code up a binary insertion sort for a Binary Search Tree, for an array.**

**Code up merge sort iteratively (again) and code up quicksort iteratively, then both recursively.**

**Code up all other basic sorts (selection, insertion, bubble) iteratively.**

**Ponder on Tarjan’s, Dijkstra’s, Bellman-Ford, Knuth-Morris-Pratt algos, and try to code them up.**

**Code up the Convex Hull algorithm. Farmer fence minimization problem, cake cutting problem ICPC**

**Work on X+Y sorting algorithm, first sort X and sort Y separately, then “join” them together.**

**Work on Frobenius number for multiple coins, write algo to compute the largest number that cannot be formed from linear combination of given numbers, computer science algorithm, think about inductive proof, strong induction, weak/mathematical induction, etc.**

**Write a chess engine, go from Greenblatt, etc., learn about board game engines, then implement a chess coach into the engines, find common mistakes in my games, themes, etc., have the engine pull up a collection of master games in this position, etc., good starting points for learning, etc.**

**For a web development project, I could make a Xiangqi (Chinese chess) site that looks as clean as chess.com/lichess, it seems that there are not that many aesthetic options. Reminds me of when I was young, there were not that many online sites, they looked like low HTML raw webpages, not that sleek or modern. Would actually be an interesting project that has some utility! Something that has not been done before. Instead of making a dogshit Amazon or Tennis Warehouse rip-off.**Diagram

Description automatically generated

**A massive project would be to make a LeetCode of hardware. First supporting VHDL, then other languages as well. Common questions, and then host hardware competitions. Have to write a VHDL synthesizer? Or am I allowed to use the open source synthesizer? I can try to write a basic synthesizer on subsets of the language for good practice.**

**Do some drone projects. Machine learning, computer vision on FPGA, or other software, to identify cars, and figure out the average length of commute.**

**Write a course selection schedule-maker algorithm, do I have access to courses in UF database? Do I have access to student count? Take into account individual preference, most important class to selector gets highest personal preference, etc.**

**Write an operating system scheduler using multiple algorithms, some heuristic algorithms, other exact optimal algorithms**

**Get a computer collection! I think it would be cool to have a history of computers, including Babbage machine diagram, vacuum tubes, Z1 and ENIAC picture, transistors, discrete components, integrated chips, Intel 4004… PDP-11, Unix source code, … would be an interesting hobby to collect these items. Preferably amongst computer scientist circle, not just buying them at auctions!**

**Explore VHDL in-depth. Try to implement a simple 2D video game with it, try to implement Tetris with it, and connect the VHDL buttons to a monitor!**

**Build flexible ALU from bare bones using VHDL. For instance, should include CLA adder, ripple-carry, hierarchical, hybrid; subtractor, multiplier (with normal, or more efficient algorithm based on bit size), divider, exponentiation, and maybe even floating point. For floating point, implement naïve division, and maybe Newton’s iterative approximation method.**

**TRY TO IMPLEMENT PROJECT EULER QUESTIONS IN VHDL, and see if the runtime on the FPGA is indeed much faster than runtime in Python, or even C++ on a general CPU. ALSO TRY DOING PROJECT EULER QUESTIONS USING PENCIL AND PAPER, CAN’T JUST RELY ON PROGRAMMING FOR EVERYTHING! Good to practice bare pencil and paper math skills.**

**You can slot multiple breadboards next to each other and connect them with wire to create more space! Try to build an entire 16-bit CPU using breadboard, wires, and integrated chips, (or even discrete components, transistors, capacitors, etc., lower than gate level if you want to go crazy)!**

**Work towards FPGA-implemented computer vision! Viola-Jones algorithm on FPGA?**

**Try to come up with an in-place merge sort! I haven’t seen any in-place merge sorts on the internet anywhere. It could be pretty novel!**

Computer vision. Computer vision. Machine learning. Start with face detection, that is a classic. Study machine learning algorithms, practice 2D cartesian plane algorithms, convex hull trick, linear regression, statistics, algorithm to group cluster of points to TWO linear y = mx + b lines, should be computationally much more difficult than just one line linear regression, since possibilities/search space with TWO lines is much more vast, can I generalize it to three lines, four lines, etc.? Image compression, Fast Fourier Transform, data compression, data filter.

Viola-Jones face detection algorithm.

Then instead of generating Boolean return value, “is human face or not”, I want to develop a classifier. Take in a bunch of image data from web, Google images, online collaborative organism classification sleuths/hobbyists that already have CORRECT species tagged to image, train the algorithm, tweak the weights until algorithm can detect which race of human, then which species of great ape, then which species of primate, then which species of mammal, which species of insect, which animal (aquatic, land, sky, maybe some clever heuristics to first see what the surroundings are, could be a caterpillar on a leaf, worm in the soil, worm under the microscope against microscope slide, artist’s rendition of organisms, then expand to plants, fungi, bacteria as well. Make the species classifier general, be able to classify all organisms. Could be difficult, i.e., how to distinguish between fungi mushroom cap, and a jellyfish? After verifying the correctness of my species classifier software, make it public, publish it online with a http host, so others can use my tool! Another option is to make it downloadable as an application, but that will require re-installations and updates. Also feasible algorithm could be simply checking photo against all photos online, which has a massive data size, probably terabytes, petabytes, exabytes, zettabytes of data! Core algorithm should only come with some important data for general classification of organisms, and primary heuristic filters, is the blue in the image water, sky, is the green grass, leaf? Should deal with different magnifications. Large panorama, close-up shot, 10 feet away, microscope? Then core algorithm should come with quick classification to see if 6 legs (insect), 8 legs (arachnid) 4 legs (vertebrate), 100 legs (centipede, caterpillar, or jellyfish (although tentacles are not as straight and structurally rigid as hexapod (insect) legs)), 0 legs (snake, worm), etc., to filter out possibilities, and then query internet/internet data bank for only those images, and then narrow the search from there, to make it more efficient. Since this is quite daunting, and very general, could be deep learning.

Computer vision on stock market trend data. Maybe computer vision, and complete the picture, can capture the intrinsic details better than a simple linear regression? Get 1 mo, 3 mo, 6 mo, 1 yr, 5 yr, max stock price history trends from Yahoo finance, then machine learning to predict where the stock price will go in the future? Base the prediction on the sector of the company, previous history of company itself, similar companies, company size/total valuation, and time period (general economic recession, growth, etc., wartime, peace time, etc.)

Medical computer vision, tumor detection.

Drone algorithm, check the commute time of cars, given a 2D bird’s eye camera view video, check the commute of cars (first, computer vision algorithm should be able to detect what is a car, vs what is a large animal moving. As the drone flies higher, the size of the car in the video would be smaller, however). Simply tracking internally the commute time for each car, and asking drivers to time their commute is an option, but drone and computer vision method would not require the dynamics and collaboration with millions of other people! My Professional Communication for Engineers end-of-semester project idea.

Figure out how Quine-McCluskey algorithm works. Try to write an automatic Karnaugh Map algorithm? Try to write a good Boolean algebra simplifier.

I think the idea of a motion detector with Xbox 360 Kinect device rigged to capture motion, as I saw from the video that had computer vision moving basketball hoop, could be also used to detect motion of tennis ball. Tennis robot should be able to mimic players, based on video. Generalizing to all different perspectives (various side views, bird’s eye, court level, TV view, ground camera view, in-person real-time video capture), would be difficult.

**TENNIS ROBOT SHOULD HAVE A COACH MODE! ANALYZE THE QUALITY OF MY SHOTS, AND ALSO ANALYZE THE WAY I AM HITTING THE BALL. HITTING IT LATE, BAD PLACEMENT, TOO SHORT, TAKEBACK IS NOT EARLY, OFF-BALANCE, ETC. Teach the robot what good balance is, by showing it videos of good quality shots vs. bad quality shots.**

Be able to analyze a player, give an assessment, to make it fun make a similar comparison to a pro player, give tips on how to improve, find weak points, strong points in a player’s game, based on break points, forehand/backhand consistency, winner-to-error rate, how deep the ball is, ball speed, spin, tactics, etc.

Machine learning algorithm to predict which direction a player is going to serve in, from just prediction based on previous results (Bayesian inference model?), to “tells” such as Agassi’s discovery of Becker serve tell seconds before serving, to serve motion, fraction of a second before contact, body position and racquet angle at contact. DO PRO PLAYERS PLAY GUESSING GAMES TO PREDICT WHERE A SERVER IS GOING TO SERVE, TO HONE THEIR ANTICIPATION?

Try to write a natural language processor. That would be interesting. ChatGPT-esque, grammar correction, plagiarism detector, etc.

For many of these ideas for applications, they would be best implemented as a background program continuously running, as opposed to a one-time thing. For instance, species classifier software should be able to take in user-uploaded photos and add them to the data collection bank.

\*\*\*First step, do MIT Intro to Python projects, final is web scraper/web crawler. Then do e-commerce. Minesweeper will be done in Programming 2.

\*\*\* Write a sudoku solver, a checkers engine, a chess engine, using classic alpha-beta pruning, my own twists and touches, machine-learning, deep learning neural network, etc.

\*\*\* Do lots of work on Alice and Bob games, board games, poker, snakes and ladders, etc.

***MINE ATP DATA! Forehands, crosscourts, points won, who’s point it is, who won the point, tennis tactics, etc.***

**Computer vision home surveillance detector, ML to detect firearms, knives, dangerous armed criminals who may be wearing masks, certain clothes, etc., detect face/body shape to determine it is a person. Maybe install raspberry PI or Arduino and a low-quality camera to transmit visual data for the algorithm to process, cheap, but effective, make sure it does not have any false positives.**

**Medical primary care automated physician using ML, instead of just training model with CT scan and MRI images a la computer-assisted radiology, maybe let the machine learn like how a human would. All the years of undergraduate science study, medical school, residency, and fellowship give humans a deep understanding of physiological functions and medical knowledge. What if I trained primary care application to be able to make probabilistic deductions/conclusions of what illness this person has, based on genetic data (openly-available genetic databank, and clinical diagnoses of diseases associated with that genetic sequencing of that individual), height, weight, age, ethnicity, etc., and symptoms. What foods this person eats, we could give a list of nutrition of certain food, tell patient to list out diet, recall any foods they eat, where they eat, how they cook, what foods they eat, and figure out if they are missing certain nutrients, low fiber, etc. So diagnosis of rare diseases can be plausible, after all, a physician can only learn so much and know so much, pick up and specialize in a limited number of topics. That is why we make MASSIVE knowledge graphs, use algorithm to assist in diagnoses of rare diseases, which often go unnoticed, and undiagnosed, or wrongly diagnosed, for a long time.**

**Work on ML algorithms, computer vision, extremely helpful in healthcare.**

**Could make a speech-to-text ML algorithm, make an application out of it, for certain words,**

**Could make coding faster, since coding is just typing, automatically includes semicolons and stuff, “code verbally,” that would be cool.**

**Also write a bunch of numerical analysis and algebraic equation solver, integral solver, write your own scientific calculators, and the algorithms to calculate common functions like large multiplications, exponentiation, logarithm, square root, powers, division, etc. That would be a cool project.**

**Also an expression simplifier, that would be cool, like LeetCode Basic Calculator IV.**

**MAKE SOME OPEN-SOURCE DATA COMPRESSION CONTRIBUTIONS !!!!!!! Data compression algorithm.**

**Keep on being creative, think outside of the box, don’t get caught up in the norm, don’t be too much of a deviant, but thinking requires deep concentration, quiet, and peace. But don’t just focus on yourself, or the things that are immediately in your vicinity. Focus on those with GREATTTT DILIGENCE of course, but in the same way looking at a small region on a chess board hinders breadth of knowledge and understanding, having narrow horizons in the real world does the same. Think BIG QUESTIONS, see what the major problems are in the world, and try to solve them. Take inspiration !!!!!!!**

**Tennis ROBOT, with Machine Learning, Physics, adjustable, based on racquet swing weight, trial and error, analytics, probability, anticipation, opponent pattern recognition, recognize how Djokovic/Federer/Nadal/Murray play, machine should be able to imitate how a particular top pro plays, as well as how I play, even imitate the types of errors I tend to make, strengths, weaknesses, etc. Then I can know how it is like to play against myself! The tennis robot should analyze videos, court level, and highlights videos from tournaments to mine the data needed to draw conclusions about tennis players. The machine should also be a UTR 18, be able to beat anyone, but be able to play at different skill levels, and able to play with different styles (consistent grinder, aggressive but less consistent). Increase temperature, wind, altitude, and humidity as factors, as well as court speed, adapt to balls, and tennis racquet, and strings, etc. Robot should not be too noisy, or else no one would want to use it commercially. The robot should also have safety features. The high speed of tennis, the swinging of a firm racquet can cause serious injury if people are in the way. Collisions are possible, so cameras on all sides of the robot should be installed as a safety feature to detect if there are nearby objects, and especially nearby people, that could create a dangerous situation. Robot would also be able to play with any racquet, and quickly adjust based on new weight, strings, and initial shots. Preferably a cheap racquet with Kevlar strings, but would that be damaging to the robot?**

**Natural language processor, analyze a particular person’s speech/text/email/formal writing patterns, and then have the AI imitate the speech style of a person. Some interesting data would be level of education, birthplace, region of residence, multilingual ability (maybe multilingual people would speak in a “hybrid” way, mixing some speech patterns of another language into a language, especially in the case of second languages), profession (people may tend to speak differently based on field of work, due to different environment and exposure – after all, people spend a lot of their time at work, 40 hours a week, that is significant), where they went for college, height, weight (body proportions may affect the overall “vibe” or character of one’s speech), ethnic differences in speech, cultural differences in speech, personality, etc., and also all of the same information for the parents as well, there could be some assimilation, children pick up what their parents do, etc.. Honestly, this project seems more difficult than the tennis robot project.**

**HOW MUCH ENVIRONMENTAL CHANGE, LEARNED BEHAVIORS, CARRY OVER INTO SPERM? IF I AM SUPER ANGRY, WILL THE NATURE OF MY SPERM BE GENETICALLY OR CHEMICALLY DIFFERENT THAN IF I AM IN A REALLY GOOD STATE OF MIND? IF I AM LAZY, WILL THE LAZINESS FEEDBACK MECHANIMS IN THE GENES BE EXPRESSED MORE IN THE SPERM? I think I should always be the best person I can possibly be, and think hard, if I am to maximize the quality of life of the next generation, even at a genetic level!!!!!!! EPIGENETICS !!!!!!!**

**I SHOULD WRITE MY ABUNDANCE OF IDEAS OUT EVERY DAY, IN THE JOURNAL ENTRIES FOLDER I RECENTLY CREATED. MY WRITING WILL IMPROVE, AND SO WILL MY THINKING !!!!!!! AS WILL MY OVERALL UNDERSTANDING OF MYSELF AND THE WORLD !!!!!!!**

####### human brain has myelin sheath, the more times a task is performed, the more efficient it becomes. TRY TO MAKE SOMETHING SIMILAR IN MACHINE LEARNING, THE HEAVILY-USED AND IMPORTANT ALGORITHMS GET THE MOST WEIGHT IN A DISTRIBUTED SYSTEM, HAVE QUICKER CONNECTIONS AND OPTIMIZATION !!!!!!! PRETTY SURE THIS IS JUST NEURAL NETWORK LOL

JOIN BOBDA’S LAB !!!!!!!

RECONNECT WITH FRANJO

MIT courseware project 5 Intro to Python news web crawler

* Maybe build an advanced graphing calculator

Then you can work on a Frontend based project like News app.

For a good resume you should try a full stack project. For example:

1. E-commerce website.
2. Blogging platform.
3. Instagram clone.
4. Resort booking app.
5. Simulation
   1. Design a single page site which contains a couple of input boxes for the user, after each input the page calculates for them a final result, this could be done for costing, performance, physics etc.
   2. Teaches you about scripting and behind the scenes code the user can’t see
6. Log in interfaces
   1. A form where a user can create an account, log in, change account, delete account and can have user privileges based on account level.
   2. Teaches you security protocols, databases, and input validation
7. A basic forum
   1. Can extend from point 2a if you want a larger project, a page where users can submit a question or their thoughts, and other users can reply
   2. Teaches databases and how to show if another user is online or not. Could be considered social networking.
8. A basic wiki-pedia style site
   1. Many many many pages, could fill them with Bacon Lorem ipsum, maybe a javascript scrolling navigation bar
   2. Teaches user end navigation, breadcrums, how to change pages, redirecting.
9. A game website
   1. This could either be a game you have made embedded in the browser using javascript, html5, flash or java. Or you could alternatively open a game from another website but it shows in your site instead.
   2. This can teach you embedding elements in a form.

Tennis robot

Church projects or non-profit work usually will check the right boxes. It usually pays at least a small amount of money, it's very forgiving resume builder work, and it'll teach you how to operate on shoestring budgets which will be a standard operational parameter for as long as you have a career in just about anything.

You can also fork(make your own sandbox copy) of just about any project or code on github. Sometimes when you're struggling, you can see what you couldn't see on your own by looking at the ways other people see or approach code.

Stack overflow is also a great place to grab a few questions and Google, research, and test code your way through some real world problems.

That's some basic where to start, but there's tons of learn at your own pace websites out there largely or entirely for free as well.

Write paper on tennis AI !

1. Develop a personal website to showcase your skills and experiences.

Try to develop clone of major sites likes

OLX, Instagram, Twitter.

And use seed project from GitHub,

Seed project is a package with good design principles, add-on plugins.

* Bonus points if you are familiar with [Command Line Interfaces](https://www.quora.com/topic/Command-Line-Interfaces), and at least one [shell scripting](https://www.quora.com/topic/Shell-Scripting) language like [Bash](https://www.quora.com/topic/Bash-Unix-shell)

1. Create a real-time To-Do list using AngularJS - very simple project
2. # with buzz notification
3. Build a calendar with the ability to add events.
4. Create a blog engine
5. Write a web application to catalog your DVD|Books|MP3 collection
6. Create a service where other developers can use that service and provide a real value to the web (how's that for open-ended?) ;-)

1. A business site/ Investment site
2. A school/hotel management system
3. A blog created from scratch by you
4. A chat web app or forum
5. E-commerce site
6. Restaurant site
7. And a couple of few simple sites should do.

105 views

View 1 upvote

* Canteen Food Ordering and Management System
* Workflow Management System for MNC
* Secured Merchant Payment using Biometric Transaction
* Online Personal Counselling
* E-Visa Processing & Follow Up System
* Loss Prevention System in Stock Market Trading
* E Commerce for Online Medicine Shopping
* Online Transaction Fraud Detection using Backlogging on E-Commerce Website
* Employee Performance Evaluation & Appraisal Calculation using Data Mining
* Online Secondhand Book Buying & Selling Portal
* College E Print Service Management
* Internet based Discussion Forum
* Increase Productivity Using Quality Management System
* Diabetes Prediction Using Data Mining
* On Road Vehicle Breakdown Assistance Finder
* Online Faculty Staff Directory for Multi University
* Customer Targeted E-Commerce
* Data Mining for Sales Prediction in Tourism Industry
* Hotel Recommendation System Based on Hybrid Recommendation Model
* Online Health Shopping Portal With Product Recommendation
* College Forums with Alumni Based on Content Filtering
* Advanced Intelligent Tourist Guide
* Industrial Visit Planning & Booking System
* Intelligent Tourist Guide
* Online Pizza Ordering System
* Personality Prediction System Through CV Analysis
* Secure Backup Software System
* TV Show Popularity Analysis Using Data Mining
* Twitter Trend Analysis Using Latent Dirichlet Allocation
* Secure E Learning Using Data Mining Techniques
* Price Negotiator Ecommerce ChatBot System
* Predicting User Behavior Through Sessions Web Mining
* Online Book Recommendation Using Collaborative Filtering
* Monitoring Suspicious Discussions On Online Forums Php
* Fake Product Review Monitoring & Removal For Genuine Ratings Php
* Detecting E Banking Phishing Using Associative Classification
* A Commodity Search System For Online Shopping Using Web Mining
* Detecting Phishing Websites Using Machine Learning
* Secure Electronic Fund Transfer Over Internet Using DES
* Sports Events Management Platform for Colleges
* Secure Online Auction System
* Filtering political sentiment in social media from textual information
* Evaluation of Academic Performance of Students with Fuzzy Logic
* Cooking Recipe Rating Based On Sentiment Analysis
* Student Information Chatbot Project
* Website Evaluation Using Opinion Mining
* Social Media Community Using Optimized Clustering Algorithm
* Preventing Phishing Attack On Voting System Using Visual Cryptography
* Search Engine Using Web Annotation
* Secure File Storage On Cloud Using Hybrid Cryptography
* E Banking Log System
* Heart Disease Prediction Project
* Stream Analysis For Career Choice Aptitude Tests
* Product Review Analysis For Genuine Rating
* Periodic Census With Graphical Representation
* Android Smart City Traveler
* Android Campus Portal With Graphical Reporting
* E Commerce Product Rating Based On Customer Review Mining
* Website Evaluation Using Opinion Mining
* Online Mobile Recharge Portal Project
* Sentiment Analysis for Product Rating
* Detecting E Banking Phishing Websites Using Associative Classification
* Opinion Mining For Social Networking Site
* Opinion Mining For Restaurant Reviews
* Monitoring Suspicious Discussions On Online Forums Using Data Mining
* Opinion Mining For Comment Sentiment Analysis
* Fake Product Review Monitoring And Removal For Genuine Online Product Reviews Using Opinion Mining
* Biomedical Data Mining For Web Page Relevance Checking
* Web Data Mining To Detect Online Spread Of Terrorism
* Smart Health Consulting Project
* Web Filtering Software
* Webpage Ranking Search Engine With Seo Suggester
* User Web Access Records Mining For Business Intelligence
* Banking Bot Project
* Intelligent Chat Bot
* Unique User Identification Across Multiple Social Networks
* Custom Web Search With User Centric Map
* Social Network Privacy Using Two Tales Of Privacy Algorithm
* Web Mining For Suspicious Keyword Prominence
* Customer Behaviour Prediction Using Web Usage Mining
* Web Content Trust Rating Prediction Using Evidence Theory
* Topic Detection Using Keyword Clustering
* An Adaptive Social Media Recommendation System
* Exam Cell Automation System
* Automated Timetable Generator Php
* College Admission Predictor Php
* College Social Network Project Php
* CRM For Internet Service Provider Php
* Customer Focused Ecommerce Site With AI Bot
* ERP System For College Management Php

Well, you asked about "computer science" instead of "programming" or "software engineering", so we need a project with some science in it.

Let's take a very simple, useful algorithm: dense [matrix multiplication.](https://en.wikipedia.org/wiki/Matrix_multiplication) It has been studied for decades and there are a lot of algorithms out there.

1) Implement several of these algorithm in multiple languages and run them using various matrix sizes on machines you have access to. You'll get some nice-looking graphs and not a lot more.

2) Using the assembly code and hardware performance counters, explain why (at an architectural level) certain algorithms are faster than others. That's well beyond what I would expect a new undergrad to be able to do.

3) Using your knowledge of the processor architecture, reason about what the fastest possible speed is for this problem --- what is the theoretical upper bound on performance? (Now you're doing computer science.)

4) Hand-tune your code (probably in assembly) to get as close as you can to your theoretical bound.

This was (roughly speaking) the project I gave an undergraduate computer science student I mentored over the summer. He was pretty sharp and made quite a bit of progress. If you can do similar work as a high school student, definitely drop me a line. I can point you to a few professors who would be interested to have you in their classes.