

Forum:[Hack Forums](#)

Thread: [How to earn a high paying software job as a new Computer Science graduate](#)

6577657534735 - December 9, 2015, 10:26 am

Light introduction to landing top developer jobs, whether you are a new grad or have been in the industry a bit. All of these are Computer Science degree fundamentals. This is also a really minimal list, there are many things you should know which realistically you won't be asked about in an interview though. Anyway:

Two obstacles:

- Getting your resume noticed
- Passing technical screenings (phone and on-site)

To get your resume noticed have a good GPA and some past internships, and if you don't have either of those make sure you have some amazing personal projects. Not little trivial ones but really cool ones that demonstrate CS fundamentals

- Design a programming language
- Write an online chat program
- Write an emulator
- Make your own high level wrapper/framework for a tedious task
- Make a game from the ground up

Employers don't care about obscure language features, or even OS or API specific knowledge. This can be learned trivially by anyone with a grasp of the fundamentals. They care about raw CS knowledge i.e. data structures and algorithms, something never ever discussed on HF.

The primary data structures are:

- Arrays
- Linked lists
- Stacks
- Binary search trees (balanced and unbalanced, RB vs AVL vs Splay)
- N-ary trees
- Heaps (min and max, often used for implementing priority queues)
- Queues (FIFO, LIFO, priority)
- Hash tables (chaining and probing collision resolution methods)
- Graphs

The primary algorithms are:

- Search, insert, and delete on all of the above data structures
- Sorting (selection and other shitty ones, then top ones which are quick sort, merge sort (online and offline), and heap sort)
- Graph traversal (breadth first search, depth first search, A*)

For all of these data structures and algorithms, Big-O notation is absolutely foundational. It's not just some dumb thing taught in one class at the beginning of a CS degree that you can forget, it is 100% relevant to every single moment you spend programming for the rest of your life and if you don't understand it, at the very least at an intuitive level if not at the mathematical proof level, then you will suffer as a developer

Furthermore it's not just reciting algorithms you've memorized it's understanding them fully, understanding the tradeoffs in different implementation methods for each data structure, and understanding the advantages of some data structures over others, both at the Big-O complexity level and the actual CPU overhead level. In some use cases a binary search tree will be faster for retrieval than a hash table despite its Big-O complexity being higher.

Know how to combine data structures i.e. "I'd use a linked list but with a hash table mapping values to linked list node pointers for $O(1)$ access and assume values are unique". Or "I'd map IDs to a binary search tree via a hash table, and each binary search tree will store the current values associated with the ID for $O(1)$ retrieval and deletion of the max and min value". Knowing what data structure is best for the job is mandatory

Then there are fundamental programming and systems concepts you need to know, the main ones are:

- A language in-depth, preferably both an interpreted and compiled one
- Memory management (garbage collection vs. malloc vs RAIL)
- Multiprocessing with both processes and threads
- Caching (both fundamentally for memoized algorithms, and both CPU memcaching and OS filesystem buffer caching)
- How high level languages map to assembly language, what stack frames are and what information they hold, the advantages of recursive vs. iterative algorithms that do the same thing
- OS concepts like how system calls work, what a kernel is
- How threads and processes are implemented in the kernel, what process scheduling is and how it's implemented internally via timer interrupts, block and run queue, etc.
- Networking: Ethernet/802.11 vs IP vs UDP/TCP vs DNS/HTTP/FTP vs HTML vs Javascript etc., from the bottom up
- Bit manipulation using masking with | and &, generating masks of desired size i.e. $(1 < n) - 1$ etc.
- Have a vague idea of how to answer any "how does it work?" question from the top to the bottom i.e. "What happens when you hit enter on a google search?" or "what happens when you press the 'b' button on the keyboard?"

Know all of these things inside and out and you can land a top job. I mean inside and out, 100% can implement any of these algorithms or explain any of these concepts off the top of your head without reference material, and also be able to explain them and combine them and discuss tradeoffs and make design decisions where the answer isn't clear. You will be asked questions where you need to design a set of data structures which work together to solve the problem, and there may be an obvious solution but that obvious solution is shit and slow, and you need to keep moving forward to find the solution with the best Big-O complexity

If you can do all of these things but also do them confidently while simultaneously knowing the limits of your knowledge, you will land a top job. But once you know these things that's still not enough, you need to:

- Apply to a shitload of companies, both top ones being optimistic but also fallback companies in case you need a backup
- Practice interviews, take every interview you can because the act of whiteboard coding is SO much different than doing it on an actual computer with access to the internet
- Write a good resume, doesn't matter if you can pass an interview if you can't even get them

That's a very light explanation of what to do. Hopefully some people will benefit from it

There is a lot of emphasis on, well, other topics on HF, mainly related to malware writing. Obscure OS specific API knowledge. And yeah those topics are good but they won't really help you get a job as a developer. But ones I listed will

Guide of the general course layout of a CS degree at most universities:

Freshmen year:

Intro to programming (Java, Python, maybe C)

Object Oriented Programming (generally Java)

Basic Algorithms and Data Structures (sorting methods, linked lists, binary search trees)

Calculus 1-2 (unless you did AP in high school then you can start at Calculus 2 or 3)

Discrete Math (proofs, number theory, algorithmic complexity analysis (Big-O notation))

Statistics (normal, student's T distributions and others, statistics tests for equality, etc.)

Sophomore year:

Digital Logic Circuit Design (designing clocks, adder/division/multiplication circuits using NAND gates and AND/OR/NOT gates, etc.)

Computer Architecture (assembly, CPU design) (assembly language mainly but a bit of CPU design for buses, conditional expressions, etc)

Calculus 3

Linear Algebra

Junior year:

More advanced Algorithms and Data Structures (hash tables, balanced binary search trees, dynamic programming)

Programming Language design (interpreted vs. compiled, static vs. dynamic, etc)

Operating Systems (how they work, interrupts for process switching and scheduling, file I/O buffer, etc.)

Theory of Computation (finite state machines, Turing machines, halting problem, P vs. NP and NP completeness)

Embedded Programming (programming an Arduino for instance and learning how to do so without helper libraries whatsoever)

Senior year:

Varies but generally lots of electives of your choice, such as:

Database design

Advanced Operating Systems (internals)

Low Level Networking Protocols (TCP/IP, UDP, HTTP, FTP, etc.)

Compiler Design

Advanced/Applied Algorithms and Data Structures

Computer and Network Security

Computer and Network Forensics

Artificial Intelligence

Natural Language Processing

Intro to Machine Learning

tl;dr

Data structures and algorithms, know them and you will make bank. This is actually the only legit "Follow this guide and you will make 100k a year" guide on HF.

Suggested books

General CS knowledge:

-Introduction to Algorithms (CLRS) - THE algorithms and data structures introductory textbook

https://en.wikipedia.org/wiki/Introduction_to_Algorithms

-The C Programming language - Book on C written by the creator of C himself. If you graduate with a degree in CS and don't know C you have failed

<http://www.amazon.com/The-Programming-Language-Brian-Kernighan/dp/0131103628>

Just look up syllabuses from CS programs at top schools and look up the topics listed, read the textbook chapters listed. You can get textbooks free online in pdf form. Here's a great example. Once again it's a thing where realistically you won't use many of these specific algorithms and concepts at a job, but the fact that you know them, are familiar with them, and have them in the back of your mind, means you will be a better algorithmic thinker. One aspect of an obscure algorithm might come in handy making a new one to solve a problem. Once again this stuff is a MINIMUM level of knowledge required, not some golden ticket. This is only the beginning.

<https://www.cs.utexas.edu/~lin/cs314h/syllabus.pdf>

<http://www.fas.harvard.edu/~libcs124/E124/syllabus.html>

Interview practice:

-Cracking the Code Interview (CtCI) - algorithm interview questions to study (NOT memorize, but actually attempt yourself to learn the concepts behind them)

<http://www.amazon.com/Cracking-Coding-Interview-Programming-Questions/dp/098478280X>

-How to Prepare Yourself for an Interview with Google

<https://www.mtu.edu/career/students/toolbox/interviews/prepare.pdf>

Classics:

https://en.wikipedia.org/wiki/The_Mythical_Man-Month

https://en.wikipedia.org/wiki/Code_Complete

https://en.wikipedia.org/wiki/Structure_and_Interpretation_of_Computer_Programs

-Operating Systems concepts

<http://codex.cs.yale.edu/avi/os-book/OS8/os8c/slide-dir/>

-Theoretical CS. This is where stuff starts to get interesting:

<http://www.amazon.com/Introduction-Theory-Computation-Michael-Sipser/dp/113318779X>

-Artificial Intelligence:

<http://aima.cs.berkeley.edu/>

tl;dr v2

Get a degree in CS from a respected university. Really nail down the data structures and algorithms. While doing so, the more useless and theoretical a topic feels to you as a beginner the more important it probably is. Study EVERYTHING. Few people can self study a full CS degree instead of just a minor subset of it. If you are reading this guide you are probably not one of them.

Also importantly for interview and job seeking tips, browse the cscareerquestions subreddit frequently and soak it in. This sub has made me tons of money with its great tips that I followed. Seeing the career success of others can also help give you motivation to study the topic itself as well:

<https://www.reddit.com/r/cscareerquestions>

C?h?e?r?r?y - December 9, 2015, 10:57 am

Bookmarked for later, thanks for this writeup!

Maxiim - December 9, 2015, 12:45 pm

I would also like to recommend "Cracking the coding interview" book, it mentions a lot of the similar things you mentioned and even more.

?Opulence - December 9, 2015, 12:48 pm

"data structures and algorithms, something never ever discussed on HF." so true haha.

I have my CS degree and came to HF to take a peek at the dark side. What I found was mainly skids, and a few legitimately intelligent people like yourself.

Good share man, thanks

6577657534735 - December 9, 2015, 1:16 pm

(Dec 9, 2015 01:45 PM)Maxiim Wrote: [\[CITE\]](#) I would also like to recommend "Cracking the coding interview" book, it mentions a lot of the similar things you mentioned and even more.

Yeah completely forgot, I'll edit that in

This book is a collection of puzzle and algorithms questions along with some systems programming and OOP design concepts questions. It's a must read for anyone preparing for interviews

Considering doing some sort of daily tip thing where I teach a new data structure and an accompanying algorithm in less than 100 words and a drawn picture, so this thread could sort of become a guide to read through for more specific tips. Would anyone be interested? Realistically this thread will be buried behind others soon though so whatever I guess lol.

Ram - December 22, 2015, 2:16 am

Best article I have ever read on HF! Great job op!

Anticipate - December 23, 2015, 11:53 pm

Good job OP. Not many people can read this and still feel like they're an advanced developer. Gives us all a reality check that we've got a long way to go. And yes, I'd be very interested in your threads teaching algorithms and data structures.

movsx - December 24, 2015, 8:55 am

(Dec 24, 2015 12:53 AM)Anticipate Wrote: [\[CITE\]](#) Good job OP. Not many people can read this and still feel like they're an advanced developer. Gives us all a reality check that we've got a long way to go. And yes, I'd be very interested in your threads teaching algorithms and data structures.

On contrary, what he mentions in his thread are requirements for the basic level developer.

Aside from that, nice thread Abbraluna.

TheAwesomeHidden - December 24, 2015, 3:09 pm

(Dec 9, 2015 02:16 PM)Abbraluna Wrote: [\[CITE\]](#) Yeah completely forgot, I'll edit that in

This book is a collection of puzzle and algorithms questions along with some systems programming and OOP design concepts questions. It's a must read for anyone preparing for interviews

Considering doing some sort of daily tip thing where I teach a new data structure and an accompanying algorithm in less than 100 words and a drawn picture, so this thread could sort of become a guide to read through for more specific tips. Would anyone be interested? Realistically this thread will be buried behind others soon though so whatever I guess lol.

Good Thread and I would be interested in learning data structures. I know the basic ones but I never had the time to learn new ones :D

Maxiim - December 26, 2015, 8:05 am

Personal opinion:

You will be wasting your own time trying to learn kids on HF some datastructs / algo. Because as you said - too few people are interested in it on hf.

It's much better if you have created a Resource-"chapter" in the thread with links to lectures and books concerning datastructs/algo, but also sites that can help you train them. For example: adventofcode.com

6577657534735 - December 26, 2015, 6:13 pm

(Dec 26, 2015 09:05 AM)Maxiim Wrote: [\[CITE\]](#) Personal opinion:

You will be wasting your own time trying to learn kids on HF some datastructs / algo. Because as you said - too few people are interested in it on hf.

It's much better if you have created a Resource-"chapter" in the thread with links to lectures and books concerning datastructs/algo, but also sites that can help you train them. For example: adventofcode.com

Yeah definitely not enough interest to justify too much more effort on my part. I will compile a list of some good textbooks though and post links to other guides similar to this.

Calcyon - December 26, 2015, 6:15 pm

You are obviously very knowledgeable and I'm sure this helped lots of users.

crazyycast13 - December 26, 2015, 10:30 pm

No mention of greedy algorithms 2/10.

But all these algorithms and things are part of the reason why I moved into a more security based rather than a computer science based job; didn't really want to work with data structures and winging things around faster.

Really know what you are applying for, and if you have any questions about the job posting requirements, just email the recruiter. Also, if you don't get a job after sending in a resume, send in a follow up email and ask how you could improve your resume, or if you land an interview but not the job, send a follow up email on how to improve your interview.

EDIT: forgot to mention I actually own that book "Cracking the coding interview". It's a good book, buy it and read through it.

6577657534735 - December 26, 2015, 11:11 pm

(Dec 26, 2015 11:30 PM)crazyycast13 Wrote: [\[CITE\]](#) No mention of greedy algorithms 2/10.

But all these algorithms and things are part of the reason why I moved into a more security based rather than a computer science based job; didn't really want to work with data structures and winging things around faster.

Really know what you are applying for, and if you have any questions about the job posting requirements, just email the recruiter. Also, if you don't get a job after sending in a resume, send in a follow up email and ask how you could improve your resume, or if you land an interview but not the job, send a follow up email on how to improve your interview.

EDIT: forgot to mention I actually own that book "Cracking the coding interview". It's a good book, buy it and read through it.

Oh yeah no explicit mention. Well many of the categories I mentioned are greedy for instance graph traversal (minimal spanning tree calculation can be done greedily via Prim's or Kruskal's Algorithms, shortest path via Dijkstra's is greedy etc.)

crazyycast13 - December 26, 2015, 11:26 pm

(Dec 27, 2015 12:11 AM)Abraluna Wrote: [\[CITE\]](#) Oh yeah no explicit mention. Well many of the categories I mentioned are greedy for instance graph traversal (minimal spanning tree calculation can be done greedily via Prim's or Kruskal's Algorithms, shortest path via Dijkstra's is greedy etc.)

I guess what I mean is you should kind of know the concept of greedy, because if you know that prims/kruskals are easy to understand.

Also, Dijkstra's algorithm is dynamic programming.

6577657534735 - December 27, 2015, 12:33 am

(Dec 27, 2015 12:26 AM)crazyycast13 Wrote: [\[CITE\]](#) I guess what I mean is you should kind of know the concept of greedy, because if you know that prims/kruskals are easy to understand.

Also, Dijkstra's algorithm is dynamic programming.

No. Dijkstra's is a greedy algorithm which sort of looks like a dynamic programming algorithm but isn't completely. It is a bit of a gray area in semantics really.

-In traditional greedy algorithms, local sub-problems are solved greedily resulting in a globally optimal solution

-In traditional dynamic programming algorithms, local sub-problems are solved optimally and recursively combined resulting in a globally optimal solution

Dijkstra's iteratively refines local sub-problems in a greedy way, with these greedy update iterations to sub-problems resulting in a globally optimal solution. It is very similar to Prim's

crazyycast13 - December 27, 2015, 12:45 am

(Dec 27, 2015 01:33 AM)Abraluna Wrote: [\[CITE\]](#) No. Dijkstra's is a greedy algorithm which sort of looks like a dynamic programming algorithm but isn't completely. It is a bit of a gray area in semantics really.

-In traditional greedy algorithms, local sub-problems are solved greedily resulting in a globally optimal solution

-In traditional dynamic programming algorithms, local sub-problems are solved optimally and recursively combined resulting in a globally optimal solution

Dijkstra's iteratively refines local sub-problems in a greedy way, with these greedy update iterations to sub-problems resulting in a globally optimal solution. It is very similar to Prim's

So have you been through a coding interview before? I've never actually gone through with one, just have done ones focused on network security and things.

6577657534735 - December 27, 2015, 2:16 am

(Dec 27, 2015 01:45 AM)crazyycast13 Wrote: [\[CITE\]](#) So have you been through a coding interview before? I've never actually gone through with one, just have done ones focused on network security and things.

Yeah they range from stupidly easy to really tough, and from quasi-casual conversations about concepts and design to writing actual code on the whiteboard

For most companies you have a 30 minute casual conversation phone screen to make sure you're not completely retarded and ask where your interests lie etc. Then a 1 hour technical phone interview where you type code online and they can see what you type etc. These will be based around algorithms/data structures and tend to be pretty simple to fit in the 1 hour time limit but always have a range of potential solutions from stupidly easy and obvious ones to the more perfect solutions with optimal big-O complexity which is what they are looking for

Then if you pass this the company flies you to their location, wines and dines you with a free hotel room for a night or two, and you go into their headquarters for an all-day interview torture session. Normally 3-6 interviews about an hour each. This is algorithm concept discussion and whiteboard coding. Also sort of discussing team work and problem resolution methods and overall software architecture/design principles. They scope you out and see how dumb you are and try to prod at your weak points. Once you prove you're not dumb and are a good teamwork person they step it up and give you harder questions and try to see the limits of your knowledge.

I've interviewed everywhere from random California startups to a few of the "Big-4" (Facebook/Google/Microsoft/Amazon). Those ones really hold back no expenses on the rental cars and nice hotels bit. Managed to get a few brix shitting offers. So excited

Overall though interviewing is so fun I kind of miss it. They just call you up and are like "Hey we'd like to fly you out here for an interview" and it's like hell yeah. Free trip to a new cool location, rental car if you're lucky, chill in the hotel and order room service for dinner while watching tv in a thick cotton bath robe. Go get an expensive dinner the next night after your interview and get a nice glass of wine. And not spend a penny of your own money. Fun. But it's not fun and games you really have to study hard ahead of time in case you're asked a really random question. Hardcore studying on the plane and brushing up on every random detail you can the week before

crazyycast13 - December 27, 2015, 2:27 am

(Dec 27, 2015 03:16 AM)Abraluna Wrote: [\[CITE\]](#) Yeah they range from stupidly easy to really tough, and from quasi-casual conversations about concepts and design to writing actual code on the whiteboard

For most companies you have a 30 minute casual conversation phone screen to make sure you're not completely retarded and ask where your interests lie etc. Then a 1 hour technical phone interview where you type code online and they can see what you type etc. These will be based around algorithms/data structures and tend to be pretty simple to fit in the 1 hour time limit but always have a range of potential solutions from stupidly easy and obvious ones to the more perfect solutions with optimal big-O complexity which is what they are looking for

Then if you pass this the company flies you to their location, wines and dines you with a free hotel room for a night or two, and you go into their headquarters for an all-day interview torture session. Normally 3-6 interviews about an hour each. This is algorithm concept discussion and whiteboard coding. Also sort of discussing team work and problem resolution methods and overall software architecture/design principles. They scope you out and see how dumb you are and try to prod at your weak points. Once you prove you're not dumb and are a good teamwork person they step it up and give you harder questions and try to see the limits of your knowledge.

I've interviewed everywhere from random California startups to a few of the "Big-4" (Facebook/Google/Microsoft/Amazon). Those ones really hold back no expenses on the rental cars and nice hotels bit. Managed to get a few brix shitting offers. So excited

Overall though interviewing is so fun I kind of miss it. They just call you up and are like "Hey we'd like to fly you out here for an interview" and it's like hell yeah. Free trip to a new cool location, rental

car if you're lucky, chill in the hotel and order room service for dinner while watching tv in a thick cotton bath robe. Go get an expensive dinner the next night after your interview and get a nice glass of wine. And not spend a penny of your own money. Fun. But it's not fun and games you really have to study hard ahead of time in case you're asked a really random question. Hardcore studying on the plane and brushing up on every random detail you can the week before

Sounds cool but absolutely horrible at the same time, the last technical interview I did almost killed me, couldn't imagine doing another 4 hours on top of it. How long have you been in your current position.

6577657534735 - December 27, 2015, 2:43 am

(Dec 27, 2015 03:27 AM)crazyycast13 Wrote: [\[CITE\]](#) Sounds cool but absolutely horrible at the same time, the last technical interview I did almost killed me, couldn't imagine doing another 4 hours on top of it. How long have you been in your current position.

Yeah they're fine but after a few hours you want to go home and cry at your ego and sense of self worth being ripped apart by some smug cunt asking you some random ass algorithm or trick you don't know and can't possibly invent on the spot lol.

Starts once I graduate soon

6577657534735 - December 27, 2015, 11:42 am

Discrete math really matters but not in the way you'd think. There is zero chance you'll ever be asked to prove anything on a whiteboard but discrete math knowledge can be necessary for big-O analysis.

Classes should focus more on the logic proof side of it, proving propositional expressions are equivalent and things like that. The number theory side is never really covered in discrete math classes. Sets become sort of an extension of propositional logic but regarding set membership and are fairly straightforward, and abstract algebra isn't ever really covered in these intro to discrete math classes either (I have no idea what it is at least lol). What you said you learned should have just been like the first week or two then it should've been very extensively added upon with proofs. I guess your class wasn't really rigorous unfortunately, maybe find a discrete math syllabus for CMU/MIT or something and study it all or choose a nice subset

Anticipate - December 27, 2015, 2:34 pm

(Dec 27, 2015 03:43 AM)Abraluna Wrote: [\[CITE\]](#) Yeah they're fine but after a few hours you want to go home and cry at your ego and sense of self worth being ripped apart by some smug cunt asking you some random ass algorithm or trick you don't know and can't possibly invent on the spot lol.

Starts once I graduate soon

You haven't graduated yet?! ?

You're well ahead of pretty much everyone I've ever come across at the undergraduate level. I'm impressed.

Hosea - December 27, 2015, 10:51 pm

Copy/Pasted this text into a word doc to read later :) I'm just a rookie but want to get into programming again.

6577657534735 - January 1, 2016, 11:23 am

It's the perfect time right now to prepare to start studying and applying for summer internships. You've already missed the fall session if you are reading this where most hiring takes place but this next month or two is basically the absolute 100% final cutoff

No matter who you are, if you are in school for CS or have self taught coding skills, there is no reason to not go apply for some internships right now. You might get one and get an all expenses paid 3 month vacation to a new location where you just have to code during the day and have fun at night (to put it very bluntly). You might make bank too, internships can pay well often \$30+/hr

Go apply

xenko - January 1, 2016, 11:46 am

One thing I have to say to new people: Don't forget the power of networking. Make friends wherever you can, whenever you can. Make sure you always remind people, without coming off needy, you are always learning and looking for new opportunities to broaden your horizons.

One more thing: Contribute to open-source projects. This is a very good thing to have on a resume when trying to stand out. It's often looked over by prospective employees too.

6577657534735 - January 5, 2016, 10:05 pm

Yeah networking is also very important but I personally have gotten all of my jobs through random applications. So whatever works for you

Impoofpooh - January 6, 2016, 12:23 pm

Appreciate this post, Very well done. Thanks you for including the links and I've bookmarked this for future reference.

Wish more post in this section were similar to this in quality.

6577657534735 - January 20, 2016, 11:37 pm

Getting near the end of the window for applying for Summer 2016 internships and jobs

Everyone get on it

Packets395 - January 26, 2016, 6:24 pm

Really impressive thread, found it to be very helpful. Especially the book list.

RottenTomatoes - January 27, 2016, 10:09 pm

Wow a guide on HF that actually dispenses useful career info, what a gem. Thanks for sharing.

And ps. to future applicants, please don't pass up your university career fairs. Especially if you don't have much prior experience, your success rate will be much higher if you show your resume to a real person rather than dumping it in the black hole that is an online application page.

Bulgarian - February 2, 2016, 9:58 am

Bookmarked for later, this is way more than just how to land a top job homie, thanks for the contribution!

Reading the C programming language right now, in hardcopy too! I can find any CS book, it's great!

Rockchuck - February 3, 2016, 7:52 pm

Thanks saving this for later.

6577657534735 - March 23, 2016, 11:37 pm

Your chance to apply to summer CS internships is slowly disappearing. Act now before they're all gone. It's not even hard just go apply to some

74mb0r1n3 - March 29, 2016, 5:31 am

Bookmarking this thread, thank you very much.

SIM-2939849 - March 30, 2016, 6:44 am

One of the best article I have read. Keep it Up

CanisMajoris - March 30, 2016, 6:48 am

Props to OP, as a senior in college going for a CS degree and having friends who have graduated and worked for some of the top companies in the area, this is it, this guide is it!

Doctor Fate - March 30, 2016, 4:18 pm

Great job OP this is actually one of the more HQ threads on data structures well done!

shmoz - April 13, 2016, 5:45 pm

How difficult do you think it would be for a Canadian attending an average school in Canada to eventually get interviews at top tech companies in (both in Canada and the States)?

I'm currently a first year premed but I'm going to be switching over to computer science for next year. I have a 3.7GPA, gonna be aiming for higher in the future. My school has a fairly good co-op program so I'll probably (or hopefully) be able to get 4 work placements over the course of my degree.

How good a position would that put me in for when I start applying for jobs?

6577657534735 - April 13, 2016, 6:31 pm

(Apr 13, 2016 05:45 PM)shmoz Wrote: [\[CITE\]](#) How difficult do you think it would be for a Canadian attending an average school in Canada to eventually get interviews at top tech companies in (both in Canada and the States)?

I'm currently a first year premed but I'm going to be switching over to computer science for next year. I have a 3.7GPA, gonna be aiming for higher in the future. My school has a fairly good co-op program so I'll probably (or hopefully) be able to get 4 work placements over the course of my degree.

How good a position would that put me in for when I start applying for jobs?

I have no idea what the international situation is like to get jobs in the US. It may be hard for you to get a position in the US or it may be easy but I literally can't tell you. Try /r/cscareerquestions. Big companies it's definitely easier than small ones to move internationally

As for Canada, you can probably get a job fine but not as easy in the US. And expect to earn significantly less if you stay in Canada.

For instance, Google HQ workers earn way more than Google Toronto workers even though the difficulty of passing the interviews is the same. Move to the US honestly you'll make way more.

Anyway since you're from an average school you need a really awesome GPA, 3.7 is probably okay but work on increasing it still, no reason not to. But essentially the whole guide still applies to you. Personal programming projects + algorithm interview concepts are more important than GPA once you've gotten the interview.

I don't know much about your work placements but I'd try to get internships at US companies as well, definitely apply to Google/Amazon/Microsoft/Facebook etc.

throwaway69 - April 13, 2016, 6:49 pm

Great write-up, thanks a bunch! Saving for later.

Infinity. - April 14, 2016, 7:13 am

I've been reading this book for Algorithms and it's quite awesome.

http://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844/ref=sr_1_1?ie=UTF8&qid=1460643174&sr=8-1&keywords=introduction+to+algorithms

6577657534735 - April 18, 2016, 5:14 pm

Yeah that's the best book on algorithms.

Summit1g - April 18, 2016, 5:26 pm

I'm about to fail my computer science class.

Example of what we go over:

(1) Write the toString method of the class Card. Look at the output to understand what to return.

Test it by creating a card object, and printing it.

(2) Write the subroutine createDeck that creates an array of 52 cards (one of each kind). Use the array suitArray to get all four suits.

After calling it from the main method, and printing the array, your output should be the same as the one provided.

(3) Write the subroutine exchange that exchanges two cards in an array of cards.

(4) Write the subroutine shuffle that performs random exchanges in an array of cards. It is summarized as follows.

```
for i = 0 to deck.length - 1
```

```
    r = random integer between 0 and deck.length - 1 (use Math.random())
```

```
    exchange the cards at i and r
```

Print the array after shuffling, and compare your output to the one provided.

(5) Write the method compareTo that compares cards based on value. Use the table we wrote in class as a guideline.

Sort the array, print it, and compare your output to the one provided.

homework

(6) Write the method highestValue that compares two cards c1 and c2 and returns the card that has the highest value, using the compareTo method.

It returns null if the two cards do not have the same suit.

Probably - April 18, 2016, 5:53 pm

(Apr 18, 2016 05:26 PM)Summit1g Wrote: [\[CITE\]](#) I'm about to fail my computer science class.

Example of what we go over:

(1) Write the toString method of the class Card. Look at the output to understand what to return.

Test it by creating a card object, and printing it.

(2) Write the subroutine createDeck that creates an array of 52 cards (one of each kind). Use the array suitArray to get all four suits.

After calling it from the main method, and printing the array, your output should be the same as the one provided.

(3) Write the subroutine exchange that exchanges two cards in an array of cards.

(4) Write the subroutine shuffle that performs random exchanges in an array of cards. It is

summarized as follows.

```
for i = 0 to deck.length - 1
```

```
    r = random integer between 0 and deck.length - 1 (use Math.random())
```

```
    exchange the cards at i and r
```

Print the array after shuffling, and compare your output to the one provided.

(5) Write the method `compareTo` that compares cards based on value. Use the table we wrote in class as a guideline.

Sort the array, print it, and compare your output to the one provided.

homework

(6) Write the method `highestValue` that compares two cards `c1` and `c2` and returns the card that has the highest value, using the `compareTo` method.

It returns null if the two cards do not have the same suit.

This isn't that hard. Seems like what we did in my Intro to Java courses. If you're failing this, I'd reconsider your major or start putting a lot of effort into CS. This should be the easiest course of your CS courses.

moate - April 19, 2016, 12:41 am

Fabulous topic , thank you

Bookmarked

Artificial Sun - April 19, 2016, 12:49 am

[MIT 6.001 Structure and Interpretation, 1986](#)

Old but still very relevant for students taking Algorithms and Datastructures course

Summit1g - April 20, 2016, 4:08 pm

(Apr 18, 2016 05:53 PM)Probably Wrote: [\[CITE\]](#) This isn't that hard. Seems like what we did in my Intro to Java courses. If you're failing this, I'd reconsider your major or start putting a lot of effort into CS. This should be the easiest course of your CS courses.

I'm not giving up on CS lmao.

Probably - April 20, 2016, 7:24 pm

(Apr 20, 2016 04:08 PM)Summit1g Wrote: [\[CITE\]](#) I'm not giving up on CS lmao.

Then you better put a lot of work into it over the Summer.

Yin Yang - April 20, 2016, 7:29 pm

I would also suggest actually making a website/application and now how it works, inside and out. For example, when an interviewer says, what have you done with Java? And you respond by saying it was a class I took for my undergrad, you won't get the job guaranteed. I graduated in CS. I do QA. I do manual testing along with automation testing using Selenium web driver using Java. QA is interesting and my dad's friend told me about it. I started off with getting an internship with QA while being in school. Now I'm in a full-time role. QA pays good. But just like with any position, if you're in a junior-senior level, it pays really good. It's an easy start and you don't have to have a CS major either. I have 3 years under my belt and enjoy every second. I have done outside work and have achieved certificates. Simply reading and remembering terms and definitions won't get you anywhere. You have to practice it and apply it.

6577657534735 - August 30, 2016, 12:12 am

(Apr 20, 2016 07:29 PM)Yin Yang Wrote: [\[CITE\]](#) I would also suggest actually making a website/application and now how it works, inside and out. For example, when an interviewer says, what have you done with Java? And you respond by saying it was a class I took for my undergrad, you won't get the job guaranteed. I graduated in CS. I do QA. I do manual testing along with

automation testing using Selenium web driver using Java. QA is interesting and my dad's friend told me about it. I started off with getting an internship with QA while being in school. Now I'm in a full-time role. QA pays good. But just like with any position, if you're in a junior-senior level, it pays really good. It's an easy start and you don't have to have a CS major either. I have 3 years under my belt and enjoy every second. I have done outside work and have achieved certificates. Simply reading and remembering terms and definitions won't get you anywhere. You have to practice it and apply it.

Yep always apply it, always have outside projects

Intro classes at school are always a joke, nobody takes them seriously because they know the curriculum is retardedly easy.

6577657534735 - September 22, 2016, 7:22 pm

This is a good time to start applying for internships if you're in university majoring in CS right now! It's never too early, I'm dead serious

YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW

Probably - September 22, 2016, 7:32 pm

Great thread. I've linked this to a few people before (hopefully I haven't posted on here already lol).

(Sep 22, 2016 07:22 PM)Abraluna Wrote: [\[CITE\]](#) This is a good time to start applying for internships if you're in university majoring in CS right now! It's never too early, I'm dead serious

YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
YOU NEED TO START APPLYING TO SUMMER 2017 CS INTERNSHIPS RIGHT NOW
Any tips on applying for them? Where to look for them/etc?

6577657534735 - September 22, 2016, 8:35 pm

(Sep 22, 2016 07:32 PM)Probably Wrote: [\[CITE\]](#) Great thread. I've linked this to a few people before (hopefully I haven't posted on here already lol).

Any tips on applying for them? Where to look for them/etc?

If you're new, an easy thing to do is literally just search every job board you can for "software internships": Glassdoor, Indeed, LinkedIn, Internships.com, Jobs.com

Get in a habit of literally applying to every internship you can find, it can't hurt and it takes like 5 minutes. Just spend like 10 hours one weekend applying to internships, you can bust out like 50 applications if you get the process down well. Have an excel spreadsheet to keep track

Firstly though, make sure you're skilled and have a good resume to explain why they should hire you.

Go to university career fairs too. Make sure to apply EVERYWHERE even companies where you think you have a small chance. It takes no time to apply and you might get a few bites.

Eagle 95 - September 23, 2016, 11:11 am

The knowledge plan you've described is just as we do here (with minor changes, e.g. we did algebra in freshman year, statistics in sophomore, etc.). For someone who's beginning junior year of computer science, this thread means a lot to me. Thanks for taking time to write this.

shashwat97 - September 24, 2016, 9:52 am

IMO every CS student should be doing competitive programming (codeforces, hackerrank, hackerearth, topcode, codechef, etc.) and trying for ACM ICPC. The concepts learnt while preparing are often the ones that get asked during the interviews.

SanctualProductions - September 24, 2016, 10:58 pm

Love this thread I have sent you a PM I really need some advice/help.

6577657534735 - September 24, 2016, 11:45 pm

(Sep 24, 2016 10:58 PM)SanctualProductions Wrote: [\[CITE\]](#) Love this thread I have sent you a PM I really need some advice/help.

Hey you sent me a PM, but your PM inbox is full so I can't respond to it lolol

SanctualProductions has reached their private message quota so your message could not be sent.

Here's my response though:

Computer Science is the degree to get to work in the computer field. All others pay way less and are more boring imo, but you personally might like others too like being a sysadmin or something. I don't know much about specific job availability outside of software engineering obs

It doesn't matter if you think you're bad at math, the type of math taught in high school is not the same type of math used for Computer Science.

Teach yourself a bit of python it's an easy introductory language, to know for sure whether you like programming

It's never too late to start learning, and most people start computer science degrees knowing literally nothing about programming, and that's totally fine you can still be good if you work hard

6577657534735 - October 3, 2016, 10:41 pm

Friendly reminder to start applying to Summer 2017 software internships NOW, NOW, NOW. Location legitimately does not matter, all decent internships will pay you more than enough to rent an apartment for the summer.

Here I'll make it easy for you:

<http://www.indeed.com/jobs?q=software+internship&l=>

<https://www.looksharp.com/s/software-internships>

<https://www.linkedin.com/jobs/software-development-intern-jobs>

<https://www.glassdoor.com/Job/jobs.htm?suggestCount=0&suggestChosen=false&clickSource=search>

[chBtn&typedKeyword=software+intern&sc.keyword=software+intern&locT=&locId=](#)

<http://www.internships.com/search/posts?Keywords=software+intern&Location=&Radius=Twenty&Company=&ListingType=Internship&Sort=MostRecent&FilterBy=&Page=1>

Click these links, take a look and see if anything seems interesting. Then just apply. It literally takes 5 minutes. The worst that can happen is that you waste a few hours looking at job postings and then get a couple phone interview offers. Maybe even a job

Copywriter - October 3, 2016, 10:56 pm

Posting to check up on this in the future, thanks!

LaFerrari - October 3, 2016, 11:05 pm

Taking computer science right now. Posting for later thanks.

6577657534735 - October 5, 2016, 12:24 am

Very important tips here as well, for some reason won't let me add to OP.

<http://www.kalzumeus.com/2012/01/23/salary-negotiation/>

Summit1g - October 5, 2016, 1:22 pm

I failed my first CS data structures test.

Can you give me help on Queues, Array Lists, and Linked Lists? I have a test on this next week.

jamesforg - October 5, 2016, 2:31 pm

Thanks for breif tutorial :) sir thanks i m sure gonna try it

Ni??c??k - October 5, 2016, 9:30 pm

Do you have a CS degree? If so, what prestigious university did you attend while pursuing your career?

6577657534735 - October 5, 2016, 9:53 pm

(Oct 5, 2016 09:30 PM)Ni??c??k Wrote: [\[CITE\]](#) Do you have a CS degree? If so, what prestigious university did you attend while pursuing your career?

Yes

A non-prestigious university

Ni??c??k - October 5, 2016, 9:55 pm

(Oct 5, 2016 09:53 PM)Abraluna Wrote: [\[CITE\]](#) Yes

A non-prestigious university

What do you consider prestigious?

6577657534735 - October 5, 2016, 9:58 pm

(Oct 5, 2016 01:22 PM)Summit1g Wrote: [\[CITE\]](#) I failed my first CS data structures test.

Can you give me help on Queues, Array Lists, and Linked Lists? I have a test on this next week.

You have one week to google "How do queues work", "How do array lists work", and "How do linked lists work" and not stop reading the results until you understand. If you can't do that then, yeah

Queues are like a line of people waiting for food. People enter the queue in the back, and leave the

queue in the front, asynchronously which is why the queue shrinks and grows over time

Array lists are just arrays that can grow and shrinks dynamically

Linked Lists are like this

Kind of like paper chains you probably made as a kid where each ring is a node storing a value and pointers to adjacent nodes

Matt AH - October 5, 2016, 10:01 pm

Not only should you know the fundamentals, but employers love side projects that are completed in your free time. Shows you have passion for what you're doing.

Take a look at this list of project ideas a user compiled(I know it's PHP, but make it work in Java):
<https://hackforums.net/showthread.php?tid=4322241>

Very neat thread. Thanks for the share.

Probably - October 6, 2016, 10:24 am

(Oct 5, 2016 01:22 PM)Summit1g Wrote: [\[CITE\]](#) I failed my first CS data structures test.

Can you give me help on Queues, Array Lists, and Linked Lists? I have a test on this next week.
<http://interactivepython.org/runestone/static/pythonds/index.html>

They're implemented in Python but it's easy to understand. Chapter 3 goes over data structures. Abra explained them a bit too. Also use Google like he said.

Colten - October 6, 2016, 8:54 pm

Any tips for someone who is planning to major in CS but struggles with Math?

Probably - October 6, 2016, 9:01 pm

(Oct 6, 2016 08:54 PM)Colten Wrote: [\[CITE\]](#) Any tips for someone who is planning to major in CS but struggles with Math?

Go to a university that doesn't require a lot of math, lol. Idk where you're from but I know one uni that only requires Calc 1. Most unis have some place on their website where they list requirements for every major. Look them up.

If you do end up going to a uni that requires a lot of math (like mine - Calc 1-3 + three other math courses, not including discrete), then good.

Armady1 - October 6, 2016, 9:18 pm

I'm currently in my 3rd year of high school, and looking to pursue a comp. sci major (If this matters I have a 4.2 UC GPA, and I am going to finish calculus 2 by the end of highschool). Recently, I've been learning the basics of python on code academy. Would you recommend me doing anything else on top of that?

My goal is going from python to c++ and then making some programs to put on my college application to show that I have a passion for this field.

6577657534735 - October 6, 2016, 9:51 pm

(Oct 6, 2016 09:18 PM)Minerclawz Wrote: [\[CITE\]](#) I'm currently in my 3rd year of high school, and looking to pursue a comp. sci major (If this matters I have a 4.2 UC GPA, and I am going to finish calculus 2 by the the end of highschool). Recently, I've been learning the basics of python on code academy. Would you recommend me doing anything else on top of that?

My goal is going from python to c++ and then making some programs to put on my college application to show that I have a passion for this field.

You're fine OP you don't need to prepare any more. CS degrees assume you come in knowing literally nothing about programming, always, no exceptions lol

And it's good that you are going to finish Calc 2 by the end of high school, I did that too (I assume you mean AP BC Calc). Make sure you really know Calc 2 though often the university level is a bit above BC Calc depending on your high school so it could potentially be a shock, just keep it in mind

Armadyl - October 6, 2016, 9:52 pm

(Oct 6, 2016 09:51 PM)Abraluna Wrote: [\[CITE\]](#) You're fine OP you don't need to prepare any more. CS degrees assume you come in knowing literally nothing about programming, always, no exceptions lol

And it's good that you are going to finish Calc 2 by the end of high school, I did that too (I assume you mean AP BC Calc). Make sure you really know Calc 2 though often the university level is a bit above BC Calc depending on your high school so it could potentially be a shock, just keep it in mind
Yeah I'm doing AP Calc BC, but I wanted to say it in college terms.

Thanks for the advice

J. McCarthy - October 11, 2016, 7:56 am

Dang that's pretty different than my college but I already knew that lol

+Haze - October 11, 2016, 8:41 am

tl;dr - Don't be a HF skid

nick_electro - October 13, 2016, 2:25 am

very nice thank you. very informative!

Canonsx130 - November 15, 2016, 7:52 am

Let's bring this back to the top. The content in here is invaluable and more people should read this.

summerghost - November 18, 2016, 6:00 am

Should of asked it before graduating. You need to work your ass off while you are still studying, do practical work, get some certifications as well and become completely handy with all the dev tools. Only then you grab a good job.

Insurmountable - November 18, 2016, 7:36 am

Just a heads up to my Comp Sci bois, if you're going into college and having to take Calc 2, be very careful. That class has one of the highest fail rates.

`KSI - November 18, 2016, 7:45 am

Posting so i can come back and read this later.

6577657534735 - November 22, 2016, 8:42 pm

Okay you legitimately have to get started, NOW, if you want a 2017 summer internship

If you don't start applying in literally 5 minutes after reading this post it will soon be too late

GET ON IT. Good software internships you can earn \$40/hr often with free housing and food, stop wasting time on fucking HF e-books and get to work applying CS students.

Summit1g - November 24, 2016, 2:39 pm

(Nov 22, 2016 09:42 PM)Abraluna Wrote: [\[CITE\]](#) Okay you legitimately have to get started, NOW, if you want a 2017 summer internship

If you don't start applying in literally 5 minutes after reading this post it will soon be too late

GET ON IT. Good software internships you can earn \$40/hr often with free housing and food, stop wasting time on fucking HF e-books and get to work applying CS students.

I'm often scared of applying to paid internships as I fear that I don't have the required knowledge to meet the companies standard...

Any tips?

6577657534735 - December 10, 2016, 6:52 pm

(Nov 24, 2016 03:39 PM)Summit1g Wrote: [\[CITE\]](#) I'm often scared of applying to paid internships as I fear that I don't have the required knowledge to meet the companies standard...

Any tips?

Just apply, there's nothing to be afraid of

If you don't meet their standards they'll reject you and it's no big deal. If you do meet their standards you'll get an interview!

NEVER be afraid to apply to a job that you don't feel good enough for. Companies can be weird sometimes. I've gotten offers from places I thought I had NO chance at, and I've failed to get offers from places where I thought it was obvious that I was going to get an offer

Just apply, literally that's it. Apply to every company you can, don't be ashamed if you don't feel good enough yet. Apply anyway. This is actually extremely important to climbing the ladder

Feynman - December 11, 2016, 10:19 am

(Dec 10, 2016 07:52 PM)Abraluna Wrote: [\[CITE\]](#) Just apply, there's nothing to be afraid of

If you don't meet their standards they'll reject you and it's no big deal. If you do meet their standards you'll get an interview!

NEVER be afraid to apply to a job that you don't feel good enough for. Companies can be weird sometimes. I've gotten offers from places I thought I had NO chance at, and I've failed to get offers from places where I thought it was obvious that I was going to get an offer

Just apply, literally that's it. Apply to every company you can, don't be ashamed if you don't feel good enough yet. Apply anyway. This is actually extremely important to climbing the ladder

Furthermore, some companies list their ideal candidate requirements as minimum requirements as it scares away people who are timid and unmotivated.

hazoristics - December 22, 2016, 1:58 am

This thread is long but it is worth reading

6577657534735 - December 29, 2016, 8:33 pm

Almost time for 2017

GUYS. If you haven't already started applying for Summer 2017 internships it's legitimately almost (just almost) too late so go do it NOW.

Anticipate - December 29, 2016, 9:51 pm

I posted to this thread a full year and 6 days ago. Interesting to see now how much more I understand from your post than from when I first read it. Also, thanks again for making such an insightful post and taking the time to care for others futures and success.

6577657534735 - December 30, 2016, 12:36 am

(Dec 29, 2016 10:51 PM)Anticipate Wrote: [\[CITE\]](#) I posted to this thread a full year and 6 days ago. Interesting to see now how much more I understand from your post than from when I first read it. Also, thanks again for making such an insightful post and taking the time to care for others futures and success.

No problem bro. Even if it just helps one fellow HFer make bank it's worth it

xLelouch - December 30, 2016, 5:06 am

I'm starting out software engineering degree in 3 weeks.

Is there are any advice for a starter? ofc this that are not included in OP.

Thanks a lot for instructions. it really helps

6577657534735 - December 30, 2016, 6:27 am

(Dec 30, 2016 06:06 AM)xLelouch Wrote: [\[CITE\]](#) I'm starting out software engineering degree in 3 weeks.

Is there are any advice for a starter? ofc this that are not included in OP.

Thanks a lot for instructions. it really helps

Don't get cocky and think "Lol I browse HF and program in (so and so) language so I don't have to pay attention to CS101. It's easy yeah but don't get cocky, you'll be tested on random methods and things specific to this language that are different from any languages you know, if you don't know this specific one yet.

Furthermore, start thinking about the big picture soon. Always think "what's the layer below that?" because unless you're all the way down to literal atoms and current, you can still go lower. Layer below Java? The JVM. What is JVM made from? C. What is C on top of? Assembly, through compilation. What is assembly? A human language mapping between opcode names and machine code. What's machine code? Microcode for the CPU circuitry. What's the CPU circuitry? Literally a boolean logic equation re-evaluated upon every clock cycle. How does it do this? Transistors making NAND gates. How does the way current flows cause a logic gate's input/output combinations (logic) to happen? Why does electricity flow?

Etc. Sorry went a bit autistic

Btw you've got to prepare to get autistic and OCD as fuck, thinking about EVERY edge case when designing complex programs. Every function has to accurately and safely handle every combination of inputs all the time, every object's methods have to be callable at any time, safely, in sync. The safer the edge cases are at the bottom, the easier it is to write generic, fast, safe code.

xLelouch - December 30, 2016, 4:15 pm

(Dec 30, 2016 07:27 AM)AbraLuna Wrote: [\[CITE\]](#) Furthermore, start thinking about the big picture

soon. Always think "what's the layer below that?" because unless you're all the way down to literal atoms and current, you can still go lower. Layer below Java? The JVM. What is JVM made from? C. What is C on top of? Assembly, through compilation. What is assembly? A human language mapping between opcode names and machine code. What's machine code? Microcode for the CPU circuitry. What's the CPU circuitry? Literally a boolean logic equation re-evaluated upon every clock cycle. How does it do this? Transistors making NAND gates. How does the way current flows cause a logic gate's input/output combinations (logic) to happen? Why does electricity flow? Yea, its kinda important for knowledge but why? Understanding JVM,C,ASM,Machine code is kinda useful even if ASM and machine code is normally not used in industry (without special cases ofc)but what is the point of understanding / finding about things below that?other than knowledge and curiosity ofc.

6577657534735 - December 30, 2016, 4:20 pm

(Dec 30, 2016 05:15 PM)xLelouch Wrote: [\[CITE\]](#) Yea, its kinda important for knowledge but why? Understanding JVM,C,ASM,Machine code is kinda useful even if ASM and machine code is normally not used in industry (without special cases ofc)but what is the point of understanding / finding about things below that?other than knowledge and curiosity ofc.

You have to know for sure what's going on at the bottom layers to be able to optimize efficiently and reason about what your code is actually doing on the CPU, because the only thing that affects performance is compiler optimization, algorithmic complexity, and the number of instructions executed (you want to minimize them)

You can't do that in some high level language where you don't understand the layer below. You can't do that without understanding all of the different programming paradigms and how they are converted to machine code. It's important for sure

Actuary0671 - December 30, 2016, 10:40 pm

Thanks for this! I am pursuing a minor in CS, have not finished yet, but it is very clear that data structures are so important in the field.

setsuna - December 30, 2016, 10:45 pm

Looking at the layout of the courses in the post and what I am actually taking makes me cringe. I have the most ridiculous requirements at my uni lol. So far I only had a python/c++ course. Last semester I was writing research papers on instant ramen and memorizing useless astronomy followed by an intro to engineering class that was absolutely useless. rip my education.

On the bright side this is a great post and thanks for the advice :)

Anticipate - December 30, 2016, 10:48 pm

(Dec 30, 2016 11:45 PM)setsuna Wrote: [\[CITE\]](#) Looking at the layout of the courses in the post and what I am actually taking makes me cringe. I have the most ridiculous requirements at my uni lol. So far I only had a python/c++ course. Last semester I was writing research papers on instant ramen and memorizing useless astronomy followed by an intro to engineering class that was absolutely useless. rip my education.

On the bright side this is a great post and thanks for the advice :)

Hey, don't worry man. I still feel the same way. I would give anything if my schedule was able to look like that. Unfortunately my first two years consisted of pointless gen eds and only about half the classes in the OP I was able to take, nevermind learn anything extensively. Would love to go back.

6577657534735 - December 31, 2016, 12:13 am

(Dec 30, 2016 11:48 PM)Anticipate Wrote: [\[CITE\]](#) Hey, don't worry man. I still feel the same way. I would give anything if my schedule was able to look like that. Unfortunately my first two years

consisted of pointless gen eds and only about half the classes in the OP I was able to take, nevermind learn anything extensively. Would love to go back.

That really sucks and my school was similar, a lot of this was skipped for me because they only offered the class every other semester for example, and I had schedule conflicts. No school is perfect

But PLEASE keep up with these topics on your own time. If your school is shit guess what, that means you have lots of free time once you complete your easy babby's first level assignments to do real studying on the important topics

They're not even hard if you keep thinking about the system of all pieces of CS working as a whole

Richard0101010101 - January 5, 2017, 1:13 pm

Lol, this is good but HF is more of a IT forum. These are the people that are going to make \$40k in their jobs probably fixing people's routers lol. Where the real niggas at?

6577657534735 - January 5, 2017, 9:17 pm

Well hopefully this thread can convince people not to go into something like IT

6577657534735 - March 1, 2017, 10:03 pm

Too late for you to apply to a 2017 summer internship if you're a CS student right now - YOU FAILED

But make sure not to fail next year anons. Start researching companies to apply to NOW so it's ez-pz to send in the applications when August rolls around

Summit1g - March 3, 2017, 6:51 pm

(Mar 1, 2017 11:03 PM)Abraluna Wrote: [\[CITE\]](#) Too late for you to apply to a 2017 summer internship if you're a CS student right now - YOU FAILED

But make sure not to fail next year anons. Start researching companies to apply to NOW so it's ez-pz to send in the applications when August rolls around

What do you mean we failed?

And what do you mean so its ez-pz?

6577657534735 - March 3, 2017, 7:04 pm

(Mar 3, 2017 07:51 PM)Summit1g Wrote: [\[CITE\]](#) What do you mean we failed?

And what do you mean so its ez-pz?

I mean it's too late now so you've lost your chance

And it's easy because it takes literally 10 minutes to send in an application

Summit1g - March 3, 2017, 7:11 pm

(Mar 3, 2017 08:04 PM)Abraluna Wrote: [\[CITE\]](#) I mean it's too late now so you've lost your chance

And it's easy because it takes literally 10 minutes to send in an application

I applied to microsoft, google, bloomberg, yelp, facebook, and i heard nothing.

6577657534735 - March 4, 2017, 11:09 am

(Mar 3, 2017 08:11 PM)Summit1g Wrote: [\[CITE\]](#) I applied to microsoft, google, bloomberg, yelp, facebook, and i heard nothing.

That's great, but most of those are really picky hard to get into places. I never heard anything from them the first years I applied either. I suggest you apply to at least 100 companies next time. There's no reason not to, it just takes a weekend of applying all day long instead of HF'ing ?

lazyme - March 4, 2017, 12:11 pm

(Mar 3, 2017 08:11 PM)Summit1g Wrote: [\[CITE\]](#) I applied to microsoft, google, bloomberg, yelp, facebook, and i heard nothing.

Also helps to contact the recruiters directly on LinkedIn. That's how I landed my job at Disney.

6577657534735 - March 4, 2017, 3:19 pm

(Mar 4, 2017 01:11 PM)lazyme Wrote: [\[CITE\]](#) Also helps to contact the recruiters directly on LinkedIn. That's how I landed my job at Disney.

Nice, what's the job? I assume Disney must have some legit good ones

movsx - March 4, 2017, 5:42 pm

Most of the top companies recruit for internships around September - November.

jujuger - March 4, 2017, 5:50 pm

Sir, I'd give you +1000 rep if I could!

BTW, what is starting salary and how much does GPA matter?

Also, what are some example projects I can do?

6577657534735 - March 6, 2017, 9:38 am

(Mar 4, 2017 06:50 PM)jujuger Wrote: [\[CITE\]](#) Sir, I'd give you +1000 rep if I could!

BTW, what is starting salary and how much does GPA matter?

Also, what are some example projects I can do?

Starting salary depends entirely on the company and how qualified you are, \$40k-\$100k+

GPA matters to get interviews but side projects and whiteboard skills matter more to pass them

-Make a compiler or interpreter for a simple language

-Write a generic server

-Make a complex game

Idk use your imagination

JaneTheBizBuilder - March 8, 2017, 12:36 am

well i have a question

how do i obtain a high pay job in digital marketing firm
as a digital marketer

what is the requirement

which degrees are required

YinYang - March 8, 2017, 12:37 am

This is a really nice thread man, a lot of decent information, if anybody is really into this then I suggest going and speaking to a career adviser as well

Dood - March 8, 2017, 1:26 am

How important is knowledge of abstract data types? I'm a first year CS student, so I'm trying to plan out my 4 years right now.

ASAC Schrader - March 8, 2017, 1:26 am

Im going to college for CS, need to bookmark this. Thanks!

6577657534735 - March 8, 2017, 9:35 am

(Mar 8, 2017 01:36 AM)JaneTheBizBuilder Wrote: [\[CITE\]](#) well i have a question

how do i obtain a high pay job in digital marketing firm
as a digital marketer

what is the requirement

which degrees are required

I have no idea, I don't know anything about that

(Mar 8, 2017 02:26 AM)Dood Wrote: [\[CITE\]](#) How important is knowledge of abstract data types? I'm a first year CS student, so I'm trying to plan out my 4 years right now.

It is the most important thing, more important than everything else. I'm serious

All your interviews will mainly be "which data structures are best to solve this problem?" and "which algorithm is best here? Implement it please and explain why it's the best"

It is by far the most important thing to learn and learn well

SIM-574702 - March 8, 2017, 9:49 am

I agree with almost everything you said. One of the professors told us we don't need to know algorithms by heart. We should only know their name and complexity/situations for which they are used. After deciding which one to use we get the optimized version from internet and implement it into our project.

What do you think about this ? It seems logical to me but there might be something I'm missing.

6577657534735 - March 8, 2017, 7:06 pm

(Mar 8, 2017 10:49 AM)matemistic Wrote: [\[CITE\]](#) I agree with almost everything you said. One of the professors told us we don't need to know algorithms by heart. We should only know their name and complexity/situations for which they are used. After deciding which one to use we get the optimized version from internet and implement it into our project.

What do you think about this ? It seems logical to me but there might be something I'm missing. Your professor is wrong. You do need to know the names and algorithmic complexities, but you ALSO need to know how to implement these algorithms by heart, on a whiteboard with a dry erase marker, and over the phone in a shared coding environment. You also need to be able to explain how they work verbally.

It's really a matter of choice. Yeah you can just learn the names and complexities and rely on the internet and you'll probably still end up getting a job, but it won't be a good job at a good company earning lots of money. Or you can learn the algorithms by heart and get a good job at a good company making lots of money.

Cuddi - March 8, 2017, 11:51 pm

This was a great in-depth thread my friend, top marks to you. Was supposed to be starting a

bachelor this year but fucked myself over.

nomatsi1286 - March 11, 2017, 9:02 am

only i can share is. build your online store like e-commerce store. you know programming so you can easily understand techie stuffs

Dood - March 11, 2017, 9:20 pm

(Mar 8, 2017 10:35 AM)Abraluna Wrote: [\[CITE\]](#) I have no idea, I don't know anything about that

It is the most important thing, more important than everything else. I'm serious

All your interviews will mainly be "which data structures are best to solve this problem?" and "which algorithm is best here? Implement it please and explain why it's the best"

It is by far the most important thing to learn and learn well

OK, thanks for answering. I'm currently in an Intro to Programming in Java class and it was just kind of terrifying to read through this thread and not recognize a lot of it. Taking an Intro to Data Structures and Algorithms class next quarter though ?

6577657534735 - April 19, 2017, 10:50 pm

(Mar 11, 2017 10:20 PM)Dood Wrote: [\[CITE\]](#) OK, thanks for answering. I'm currently in an Intro to Programming in Java class and it was just kind of terrifying to read through this thread and not recognize a lot of it. Taking an Intro to Data Structures and Algorithms class next quarter though ?
Late reply lol, nice and don't worry. Just make sure you study and keep on track and know it all and you will be successful

Also make sure not to acquire snobby programming biases and preferences like some people do, "I hate so and so language it's shit"

Some languages are legitimately shit, but you should know them enough to know why they're shit, and know how to hypothetically make them better etc. Because you might some day have to work on code written in a shit language, and you have to know how to avoid its shittiness ?

6577657534735 - July 30, 2017, 8:02 pm

Friendly reminder that it's now officially way too late to get a Summer 2017 software internship. So...If you just graduated high school and are about to enter your freshmen year of university, this is the perfect time to start looking for a basic Summer 2018 internship. But if you don't get one it's not the end of the world, but you should DEFINITELY try. Most people don't, but if you're a high achiever, do it.

If you just finished your freshmen year of university, make sure to get one for Summer 2018
If you just finished your sophomore year of university, you fucked up BAD and you definitely have to get an internship ready for Summer 2018, you absolutely have to period
If you just finished your junior year of university, you're basically fucked, internships don't exist for people who have already graduated so better luck next time anon. But now it's time to search for a full time job so you better start EARLY because you get LAST PICK of companies since you have no internship

Good luck, be proactive, don't procrastinate on this it will cost you millions of dollars over your lifetime, this is no exaggeration. Ideally you should have at least one internship during university, but two is amazing, three is extra amazing. Zero is unacceptable and you will have a really hard time finding a well paying full time job after you graduate

Novice Hacker - July 31, 2017, 7:11 pm

Terrific, it's also important to understand how much help a good reference can be. Make sure to connect really well to at least one teacher in your tedious yet worthwhile 4 year adventure.

exploid2017 - August 4, 2017, 12:47 am

Best article I have ever read on HF! Great job bro

Ronnie Coleman - August 5, 2017, 12:03 am

Doing competitive programming will make learning all those DS and algorithms very challenging and enjoyable. And it will allow you to think/solve/identify CS problems very quickly than average.

Quentin Coldwater - August 11, 2017, 3:51 am

Good job. You look like you know what you're talking about. Keep it up man!

6577657534735 - August 20, 2017, 9:01 pm

Great time to start looking now right at the beginning of the school year. Make sure to have your resume set up and ready to send out when companies start taking applications. Be proactive.

BismiLLah1 - August 24, 2017, 11:42 pm

To quote Google's Senior Vice President of People Operations Laszlo Bock from his New York Times interview In Head-Hunting, Big Data May Not Be Such a Big Deal:

One of the things we've seen from all our data crunching is that G.P.A.'s are worthless as a criteria for hiring, and test scores are worthless — no correlation at all except for brand-new college grads, where there's a slight correlation.

OTOH, better software engineering aptitude can make millions and even billions of dollars worth of difference to the bottom line.

This isn't to say you'll get a high-paying job without working hard on your computer science classes - just that the average grade from a degree program doesn't matter.

You need a working knowledge of data structures which comes from actually implementing them. Solutions to many interesting problems benefit from computer architecture knowledge. Success on real projects done for school (build a compiler for compiler construction, team up with two other students to do the work of a full-time employee in private industry for senior project) translates into doing the same in business. Classes like those can take a lot of time and effort.

dan773 - August 25, 2017, 11:35 pm

Im going into Data Structures this semester, thank you for the info!

6577657534735 - September 20, 2017, 9:12 pm

(Aug 24, 2017 11:42 PM)BismiLLah1 Wrote: [\[CITE\]](#) To quote Google's Senior Vice President of People Operations Laszlo Bock from his New York Times interview In Head-Hunting, Big Data May Not Be Such a Big Deal:

One of the things we've seen from all our data crunching is that G.P.A.'s are worthless as a criteria for hiring, and test scores are worthless — no correlation at all except for brand-new college grads, where there's a slight correlation.

OTOH, better software engineering aptitude can make millions and even billions of dollars worth of difference to the bottom line.

This isn't to say you'll get a high-paying job without working hard on your computer science classes

- just that the average grade from a degree program doesn't matter.

You need a working knowledge of data structures which comes from actually implementing them. Solutions to many interesting problems benefit from computer architecture knowledge. Success on real projects done for school (build a compiler for compiler construction, team up with two other students to do the work of a full-time employee in private industry for senior project) translates into doing the same in business. Classes like those can take a lot of time and effort.

I don't see how knowledge of these things don't correlate to grades, that's literally what you are graded on

Dirty - September 27, 2017, 5:40 am

This is a very nice read. Bookmarked everything for a later date. Since I'm currently a beginner and hyper focused on mastering all that I will touch these subjects in the future.

2SLiCKZ - September 27, 2017, 5:47 am

This is super helpful and informative!!! Bookmarked for sure.

6577657534735 - October 25, 2017, 12:22 am

Note that the application period for 2018 Summer Internships is coming to an end quite quick, you better be applying to lots of them in the next couple of weeks if you are a CS major right now, NO EXCEPTIONS.

Forget HF, forget your hobbies for a while - CS internships can make you up to \$20k in a single summer if you get a really good one, even \$15k for an "average" good one.

ven0m - October 25, 2017, 9:48 pm

I came to this thread thinking--Great! Another bullshit steps towards CS industry. But this is one of the best-written guides one can actually follow to actually become valuable to the company.

Thank you for sharing this guide.

6577657534735 - January 21, 2018, 9:23 pm

It's almost getting too late to apply for Summer 2018 internships now - if you are a CS major and haven't yet, you have to RIGHT NOW or you will miss out. Apply even if you don't plan to work anywhere - it doesn't hurt to get interview practice and don't underestimate yourself.

EssayBuddy - January 21, 2018, 11:26 pm

Over two years old and it still seems to be helping. Good work!

Kyrus - January 24, 2018, 5:39 am

I've been looking for something like this for a long time..

Honestly been most looking for alternatives for a well paying IT Career without college

Currently starting a program called Year Up for Database Analyst where they'll place me into an internship for 6 months after 6 months of class and put me into a job starting at \$22/hr after I finish the program.

But I kinda wanna aim higher than that..

Dear OP, for someone who dropped out in junior year of high school and just got a ged instead.. how hard would it be to follow the steps on this thread for college?

I absolutely detested school. I had about 90 absences from one semester that made me go to court

ending up being the reason I dropped out. It's not like I'm not smart or anything because I was placed in many advanced classes and skipped a few grades in mathematics, but my discipline is next to nothing. And things have a hard time retaining my interest.

6577657534735 - January 24, 2018, 9:37 am

It will be hard if you have no discipline because it requires lots of studying of topics which seem useless. But you can do it if you force yourself to have discipline and try hard. It's worth it because the end result is a fun job and tons of money.

Ramsey - January 24, 2018, 9:51 am

This was incredible, down to earth and probably the most useful thing I've read on HF.

Thank you.

Kyrus - January 24, 2018, 10:04 am

(Jan 24, 2018 10:37 AM)Abraluna Wrote: [\[CITE\]](#) It will be hard if you have no discipline because it requires lots of studying of topics which seem useless. But you can do it if you force yourself to have discipline and try hard. It's worth it because the end result is a fun job and tons of money. I always feared those college classes because of my reasons

I have no confidence in myself so I am indecisive if I want to risk going to college and possibly dropping out again.. this time with loans to pay off

Would you say there's any alternative paths for me to take?

6577657534735 - January 24, 2018, 8:27 pm

(Jan 24, 2018 11:04 AM)Kyrus Wrote: [\[CITE\]](#) I always feared those college classes because of my reasons

I have no confidence in myself so I am indecisive if I want to risk going to college and possibly dropping out again.. this time with loans to pay off

Would you say there's any alternative paths for me to take?

Either you work hard and study the material in a college environment with professors to teach you and answer questions, and the threat of failing looming over your head forcing you to have motivation to finish, or you self study on your own schedule with nothing to pressure you into doing any work or studying. If you're lazy, it's obvious self studying won't work

G0RG0N-PRODUCT - January 24, 2018, 10:43 pm

Best article I have ever read on HF! Great job op!

Steve - January 24, 2018, 11:03 pm

What a coincidence. I graduated with a CS degree in December and just got a call today to set up and interview for a company I applied for. I'll most certainly be going over this by Tuesday!

K?ng - January 26, 2018, 10:53 pm

Might I add that OSSU: <https://github.com/ossu/computer-science> is a good way for the people who want to supplement their education with some self-taught material and also the ones who may not know about some fundamentals but cannot learn it at a conventional academic institution.

Also, it is free!

Pyromania - February 7, 2018, 4:40 pm

bookmarked to never look at again, well done me.

Good write up though.

Kyrus - February 9, 2018, 3:56 am

(Jan 24, 2018 09:27 PM)Abraluna Wrote: [\[CITE\]](#) Either you work hard and study the material in a college environment with professors to teach you and answer questions, and the threat of failing looming over your head forcing you to have motivation to finish, or you self study on your own schedule with nothing to pressure you into doing any work or studying. If you're lazy, it's obvious self studying won't work

Then how do I stop being lazy

6577657534735 - February 9, 2018, 9:49 pm

(Feb 9, 2018 04:56 AM)Kyrus Wrote: [\[CITE\]](#) Then how do I stop being lazy
Take control of your own mind. Do you have free will? Or are you an automaton beholden to your primitive desires? Create a mental checkpoint. A thought in your mind, an abstract one that you can recognize any time you experience it because you know it and you've memorized it and it's so deeply embedded in your consciousness that you will never forget it. A checkpoint telling you to snap out of it and enter a moment of clarity where you think "what should I be doing right now?"

When this checkpoint hits - don't even think about what you need to be doing, just go do it. Continually keep a list of things you know you need to be doing, and any time you have a moment of clarity like this, take the first thing on your list and just do it, no matter what. Learn to program yourself.

I've just had one right now - what the fuck am I doing on HF? Time to go study.

Actually, fuck it, I've been wasting a lot of time on here recently and it's distracting me. Time to take a vacation ban so I can focus on studying a bit easier. I suggest you do the same anon. I'll be back when my break ends to check if you've listened to my advice.

6577657534735 - April 26, 2018, 7:49 pm

Summer is soon approaching, if you haven't found a Summer 2018 internship yet it's too late for the good ones, but you can still search for one at smaller companies. Don't hesitate, go apply now

Kyrus - April 29, 2018, 9:49 pm

(Apr 26, 2018 07:49 PM)Abraluna Wrote: [\[CITE\]](#) Summer is soon approaching, if you haven't found a Summer 2018 internship yet it's too late for the good ones, but you can still search for one at smaller companies. Don't hesitate, go apply now

Currently enrolled as a database analytics student now. Taking web development and SQL and spreadsheet design.

Also doing graveyard valet. Just started this week.

School is 8:30am-3:30pm mon-fri
Work is 10pm-7am wed-sat

I guess I made some progress.

6577657534735 - April 29, 2018, 11:07 pm

Nice job. Now try hard to find a job that is relevant to your degree. It's good to have a side job to make money, but experience is even more important than money since the gains will compound

over time. Even a basic internship related to CS will help you get a better job sooner and give you more negotiating power in the future, and you'll end up richer overall

SLY - April 30, 2018, 3:17 am

Posting to comeback later, thanks for the info OP.

Kyrus - May 9, 2018, 3:58 pm

(Apr 29, 2018 11:07 PM)Abraluna Wrote: [\[CITE\]](#) Nice job. Now try hard to find a job that is relevant to your degree. It's good to have a side job to make money, but experience is even more important than money since the gains will compound over time. Even a basic internship related to CS will help you get a better job sooner and give you more negotiating power in the future, and you'll end up richer overall

The program I'm doing is 6 months of class and then 6 months of internship

I got to drive some famous people's Lambo and Maserati's

Anyways just an update: I feel like dying..

Dragon. - May 10, 2018, 12:51 pm

(May 9, 2018 03:58 PM)Kyrus Wrote: [\[CITE\]](#) The program I'm doing is 6 months of class and then 6 months of internship

Man I wish my school offered a Co-Op program. :/

hedge.funds - May 10, 2018, 8:04 pm

I always look back at the 'lost' opportunity of not studying my dream-course in College.

Kyrus - May 13, 2018, 11:55 am

(May 10, 2018 12:51 PM)Dragon. Wrote: [\[CITE\]](#) Man I wish my school offered a Co-Op program. :/ It's called Year Up, you should look into it.

Most of their sites have their own instructors and stuff but the one I'm attending has a partnership with a college in the area so I'm currently a Year Up student but also that college's student.

You don't pay a dime btw, you get paid instead actually.

<https://www.yearup.org>

Dood - June 6, 2018, 5:57 pm

I'm coming back to this thread after first seeing it like 2 years ago. Just wanted to say that I really appreciate you writing this up; I'm currently nearing the end of the second year of my Computer Science degree, and hope that I'm on track. I haven't really gotten to the fun stuff yet (OS, "advanced algorithms", computer architecture, etc.) yet but I feel like I've learned a lot, and can comfortably explain a lot of the stuff that you mentioned.

I know that doing so will set me back a little, but I really want to take a Distributed Systems course. The topic seems really important, and I've heard really good things about the professor. Do you think that it's a class worth taking? By doing so, I'm missing out on an Applied Machine Learning class, sadly.

d4rkj4y - June 6, 2018, 6:02 pm

(Jun 6, 2018 05:57 PM)Dood Wrote: [\[CITE\]](#) I'm coming back to this thread after first seeing it like 2 years ago. Just wanted to say that I really appreciate you writing this up; I'm currently nearing the end of the second year of my Computer Science degree, and hope that I'm on track. I haven't really gotten to the fun stuff yet (OS, "advanced algorithms", computer architecture, etc.) yet but I feel like

I've learned a lot, and can comfortable explain a lot of the stuff that you mentioned.

I know that doing so will set me back a little, but I really want to take a Distributed Systems course. The topic seems really important, and I've heard really good things about the professor. Do you think that it's a class worth taking? By doing so, I'm missing out on an Applied Machine Learning class, sadly.

You can do both, just learn them in your own time with your own resources, that is what I'm doing - more specifically to big data (including distributed systems to actually process the data), machine learning genetic algorithms etc

Dood - June 7, 2018, 9:05 am

(Jun 6, 2018 06:02 PM)d4rkj4y Wrote: [\[CITE\]](#) You can do both, just learn them in your own time with your own resources, that is what I'm doing - more specifically to big data (including distributed systems to actually process the data), machine learning genetic algorithms etc

I heard that distributed systems can't really be learned on your own, since it's so new that online resources are mostly outdated already. Don't know if it's true, but that's what I've heard.

d4rkj4y - June 7, 2018, 5:45 pm

(Jun 7, 2018 09:05 AM)Dood Wrote: [\[CITE\]](#) I heard that distributed systems can't really be learned on your own, since it's so new that online resources are mostly outdated already. Don't know if it's true, but that's what I've heard.

It really all depends on what you'd like to accomplish. Distributed systems = scaling horizontally (ie the future of technology as you know).

If you'd like to leverage what has already been build, solutions such as Hadoop YARN, apache kafka clusters, AWS cloud computing, apache ignite distributed data fabric across multiple nodes, or even utilizing DCOS datacenter management (to manage the clusters), then it's a matter of understanding the underlying logic and letting the framework do the heavy lifting for you.

Or if you're more into interacting with the machines at a raw level as an academic then I can see why you'd opt for a course.

I'm guessing that instead of re-building solid systems you're more after leveraging what is already out there.

6577657534735 - June 8, 2018, 7:55 pm

(Jun 6, 2018 05:57 PM)Dood Wrote: [\[CITE\]](#) I'm coming back to this thread after first seeing it like 2 years ago. Just wanted to say that I really appreciate you writing this up; I'm currently nearing the end of the second year of my Computer Science degree, and hope that I'm on track. I haven't really gotten to the fun stuff yet (OS, "advanced algorithms", computer architecture, etc.) yet but I feel like I've learned a lot, and can comfortable explain a lot of the stuff that you mentioned.

I know that doing so will set me back a little, but I really want to take a Distributed Systems course. The topic seems really important, and I've heard really good things about the professor. Do you think that it's a class worth taking? By doing so, I'm missing out on an Applied Machine Learning class, sadly.

If you want, post a syllabus of the distributed systems course here and I can tell you if it looks good

Distributed systems is one of those courses that can either be shit and focus on dumb stuff like APIs and specific company's cloud products and other shit like this, or it can be an excellent systems level class with important concepts. Since you're only in your second year I'm going to assume it's probably not amazing, but it could be good.

I personally would stick with Applied Machine Learning, very strongly

Dood - June 9, 2018, 9:36 pm

(Jun 8, 2018 07:55 PM)Abraluna Wrote: [\[CITE\]](#) If you want, post a syllabus of the distributed systems course here and I can tell you if it looks good

Distributed systems is one of those courses that can either be shit and focus on dumb stuff like APIs and specific company's cloud products and other shit like this, or it can be an excellent systems level class with important concepts. Since you're only in your second year I'm going to assume it's probably not amazing, but it could be good.

I personally would stick with Applied Machine Learning, very strongly
I think it's intended to be more of a low-level class, not specific to any frameworks and technologies or any of that (we do use Docker, though). Operating Systems and System Programming courses are recommended before taking this class. I don't have a specific syllabus, but I can post a bit of the topics here. Some stuff isn't mentioned, as well, since this is from an older offering by the same professor.

Quote:In each module, we dive into the middle of a subject and swim our way out, going from the abstract to the concrete:

- Abstract models
- Fundamental challenges
- Protocols and mechanisms

Module 1: Time and Asynchrony

- The Synchronous and asynchronous network models. Partial orders.
- Clock skew. Causality. Causality violations.
- FIFO, Causal and totally-ordered delivery and broadcast protocols. Lamport clocks. Vector clocks. Consistent snapshots.

Module 2: Fault Tolerance

- Fault models. Reliability. Availability. Principles of redundancy.
- Partial failure. Non-independence of faults. Replication anomalies (see also Module 3).
- Reliable delivery and broadcast. Primary/Backup, Chain and Quorum replication. Recovery.
- Write-ahead logging. Checkpointing. Failure detection.

Module 3: Consistency

- Consistency models (single-copy consistency, sequential consistency, linearizability, serializability, causal consistency and other weak models)
- Message reordering in the presence of redundancy. Race conditions. Replica divergence.
- Consistency in replicated systems. Two-phase locking. Optimistic concurrency control.
- Two-phase commit. Consensus.

Module 4: Parallelism and Scaleout

- Scaleability. Data parallelism. Embarrassing parallelism.
- Amdahl's law. CALM Theorem.
- Distributed data processing systems and programming models.
- Also some other stuff like CAP theorem, Paxos algorithms, MapReduce, etc.

KingOfSleeping - June 10, 2018, 3:11 pm

So I'm going to school for computer science in 2 years about to be a junior but my main question is do you not have to do math after your freshman and sophomore year or was it just not listed.

BagelMaster45 - June 10, 2018, 4:56 pm

Yeah I already knew this, you can make a lot of money as a software dev but MAN is it boring and fucks with your posture.

6577657534735 - June 10, 2018, 5:48 pm

(Jun 9, 2018 09:36 PM)Dood Wrote: [\[CITE\]](#) I think it's intended to be more of a low-level class, not specific to any frameworks and technologies or any of that (we do use Docker, though). Operating Systems and System Programming courses are recommended before taking this class. I don't have a specific syllabus, but I can post a bit of the topics here. Some stuff isn't mentioned, as well, since this is from an older offering by the same professor.

Also some other stuff like CAP theorem, Paxos algorithms, MapReduce, etc.

That actually sounds really good, I would highly suggest yo take it. Shame you miss out on Applied ML though, try hard to take it another way

6577657534735 - November 2, 2018, 12:06 am

It's not too late to start applying for summer 2019 internships, in fact it's the PERFECT time

If you're a CS student in your freshmen year, sophomore year, junior year, or even senior year, it's time to either start applying for internships or full time jobs (if a senior) NOW

RIGHT NOW

JUST DO IT

Because it takes 10 minutes to apply to a company. Spend a few hours, a few weekends, applying, it may make your life a lot better in the future if you get a good head start on applying

?roye - November 3, 2018, 1:29 pm

Great and Helpful thread OP, thank you for such a great share!

6577657534735 - January 10, 2019, 9:55 pm

Multi-yearly reminder that if you haven't applied for Summer 2019 internships now, it will soon become too late. APPLY NOW to every internship if you're at the end of your freshmen, sophomore, or junior year of college. There is no excuse not to try. You're missing out on lots of money, one summer internship can pay anywhere from \$5k to \$25k

Bigbop - January 22, 2019, 5:37 pm

Doesn't it all just water down to experience ? Yes. Yes it does.

6577657534735 - January 22, 2019, 7:30 pm

(Jan 22, 2019 06:37 PM)Bigbop Wrote: [\[CITE\]](#) Doesn't it all just water down to experience ? Yes. Yes it does.

Not sure what you mean. If you mean that the topics I mentioned in my OP don't matter and only experience matters, then you're only correct for very entry level jobs, not high paying jobs. Plenty of people have the same 1 year of experience repeated for 10 years, and that's very bad for your career.

k0urum - June 9, 2019, 9:44 am

Replying so I can come later, good stuff OP!

ingenious - June 11, 2019, 12:09 am

Good writeup

CTCI is such a nice book!

overlude - *June 12, 2019, 12:56 pm*

Pretty good post. I would note that for initial job-hunting, you're going to want to get good at CS-y problems (e.g. implementing algorithms, math stuff generally, lists are popular for ex). Sites like Leetcode help with this. They're usually boring but pretty much every typical tech interview is mainly made up of these Qs currently, so you'll want to be good at these problems.

mafiaboy - *July 16, 2019, 1:10 pm*

Replying so I can view this later. Great info !

ethelooking - *September 21, 2020, 9:32 pm*

I think you need open your networking when you're learning at school.