Is mathematics necessary for programming? [closed]

Asked 16 years, 2 months ago Modified 12 years, 6 months ago

Viewed 172k times

117

votes

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 12 years ago.

Locked. This question and its answers are locked because the question is off-topic but has historical significance. It is not currently accepting new answers or interactions.

I happened to debate with a friend during college days whether advanced mathematics is necessary for any veteran programmer. He used to argue fiercely against that. He said that programmers need only basic mathematical knowledge from high school or fresh year college math, no more no less, and that almost all of programming tasks can be achieved without even need for advanced math. He argued, however, that algorithms are fundamental & must-have asset for programmers.

My stance was that all computer science advances depended almost solely on mathematics advances, and therefore a thorough knowledge in mathematics would help programmers greatly when they're working with real-world challenging problems.

I still cannot settle on which side of the arguments is correct. Could you tell us your stance, from your own experience?

math

Share

edited Feb 2, 2009 at 11:05

community wiki
4 revs, 4 users 62%
simsim

7 I called up my hs math teacher ten years later just to tell him I never used advanced math my whole career. Dick move, I know, but great question! Plus one. – Chris McCall Oct 17, 2009 at 4:44

2 stackoverflow.com/questions/157354/… and many others from stackoverflow.com/search?q=should+I+study+math – dmckee --- ex-moderator kitten Mar 3, 2010 at 2:23

Please see the highest voted answer to this question.
– Boris Stitnicky Oct 26, 2013 at 2:48

Rather than. Close, this should be migrated to programmers.
– Chris Cudmore Oct 26, 2013 at 2:59

Comments disabled on deleted / locked posts / reviews

71 Answers

Sorted by: Highest score (default)

1 2 3 Next

111
votes
✓

To answer your question as it was posed I would have to say, "No, mathematics is not necessary for programming". However, as other people have suggested in this thread, I believe there is a correlation between understanding mathematics and being able to "think algorithmically". That is, to be able to think abstractly about quantity, processes, relationships and proof.

I started programming when I was about 9 years old and it would be a stretch to say I had learnt much mathematics by that stage. However, with a bit of effort I was able to understand variables, for loops, goto statements (forgive me, I was Vic 20 BASIC and I hadn't read any Dijkstra yet) and basic co-ordinate geometry to put graphics on the screen.

I eventually went on to complete an honours degree in Pure Mathematics with a minor in Computer Science. Although I focused mainly on analysis, I also studied quite a bit of discrete maths, number theory, logic and computability theory. Apart from being able to apply a few ideas from statistics, probability theory, vector analysis and linear algebra to programming, there was little maths I studied that was directly applicable to my programming during my undergraduate degree and the commercial and research programming I did afterwards.

However, I strongly believe the formal methods of thinking that mathematics demands — careful reasoning, searching for counter-examples, building axiomatic foundations, spotting connections between concepts — has been a tremendous help when I have tackled large and complex programming projects.

Consider the way athletes train for their sport. For example, footballers no doubt spend much of their training time on basic football skills. However, to improve their general fitness they might also spend time at the gym on bicycle or rowing machines, doing weights, etc.

Studying mathematics can be likened to weight-training or cross-training to improve your mental strength and stamina for programming. It is absolutely essential that you practice your basic programming skills but studying mathematics is an incredible mental work-out that improves your core analytic ability.

Share

edited May 31, 2011 at 18:15

community wiki

3 revs, 3 users 91%
Mark Reid

6 I would like to add that mathematics is not *just* weight training for your mental strength, but also opens the door to new problems and solutions. For example, during my image processing internship, there's no way I would be able to implement our noise removal algorithms without a thorough understanding of PDEs. The resulting code was simple, but very precise. And the math is what got us there. Even more amazing is the beauty of these formulas. One small PDE expression can translate to a few pages of code. Without the initial formula it's difficult to see how you'd ever end up with it. – Joe Dec 18, 2011 at 16:10

Math is not *needed* for programming, because programming *is* math. It can be good math, or bad math (like when orangutan's long call is transliterated into Ook language), but whenever a programmer designs an object model of someting, ze is (usually unknowingly) performing an act of mathematics. So, again, one doesn't need to read math books to program, but that changes nothing on the fact that programming is math, and that (as Euclid put it), there is no royal road to it. – Boris Stitnicky Oct 26, 2013 at 2:44

56
votes

While advanced mathematics may not be required for programming (unless you are programming advanced mathematics capability) the thought process of programming and mathematics are very similar. You begin with a base of known things (axioms, previously proven theories) and try to get to someplace new. You cannot skip steps. If you do skip steps, then you are required to fill in the blanks. It's a critical thought process that makes the two incredibly similar.

Also, mathematicians and programmers both think critically in the abstract. Real world things are represented by objects and variables. The ability to translate from concrete to abstract also links the two fields.

There's a very good chance that if you're good at one, you will probably be good at the other.

Share

answered Oct 1, 2008 at 12:40

community wiki
Jarrett Meyer

And so you could say that someone who does a lot of crosswords or plays Scrabble is more likely to be good at programming than someone who does not. The relationship seems to be about the likelihood rather than a strict dependency. – Liam Oct 1, 2008 at 13:22

The rigor and discipline of solving a problem in math translates to programming. You need to have the ability to deconstruct a problem and approach it from first the obvious direction, and then from more creative directions in both disciplines. – Bramha Ghosh Oct 1, 2008 at 14:08

---

4 Two great programmers I admire a lot didn't study Computer Science, but rather were trained as mathematicians and just changed career paths: Miguel de Icaza and Alexander Stepanov. By the way, Stepanov once stated that while working on math "you don't start with axioms: you *end* up with axioms". :D – Joe Pineda Oct 1, 2008 at 21:22

@Liam - It's about pattern matching and juggling things in memory in the short term. Both activities require those skills.

– jcolebrand Apr 14, 2010 at 16:23

+1 for "There's a very good chance that if you're good at one, you will probably be good at the other."
– Thorbjørn Ravn Andersen Sep 15, 2011 at 9:36

38

votes

computer science != programming

OK, seriously, I know good and bad programmers who were English and Psychology majors and some that were Computer Science majors. Some very famous guys that I admire as developers didn't have a CS background. Larry Wall(Perl), for example, was a linguist. On the other hand, it helps to know something about the domain you are working on because then you can at least see if your data makes sense and help your customer/users drill down to what they really want. And yes, there's the issue of computational complexity and efficient data structures and program correctness. That's stuff you learn in Computer Science and that's useful to know in almost any domain, but it's neither necessary nor sufficient.

Share

edited Feb 17, 2012 at 23:37

community wiki
3 revs
David Nehme

12 I think the best quote I've heard is "Computer science is no more about computers than astronomy is about telescopes."

(Dijkstra) – tloach Oct 1, 2008 at 13:11

5 I think stackoverflow.com was created basically to overcome this one-liner type of answers, yet there are 15 like-minded people who lifted up this useless answer to the top. I have nothing personal with you, however I've flagged your answer as "offensive". It is so at least to me. – simsim Oct 1, 2008 at 14:47

2 @simsim: I find it offensive that people think a CS degree is somehow about programming. – tloach Oct 1, 2008 at 14:50

3 @simsim: It doesn't matter what Joel and Jeff had in mind when they created this website. It's all about creating a community where programmers help each other. I think David's short answer is right on target. +1 – Scottie T Oct 1, 2008 at 15:41

@ Scottie: if it was "all about creating a community where programmers help each other", then forums & mailing lists would have sufficed. Maybe I was a bit harsh in my response, but it was surprising to see 15 users vote up for a completely clueless "answer", well, before editing of course. :-) – simsim Oct 1, 2008 at 21:23

28 votes

I guess I am going to be the first person to say you **do** need math. As others have said math is not all that important for certain aspects of development, but the fundamentals of critical thinking and structured analysis are very important.

More so, math is important in understanding a lot of the fundamentals that go into things like schedulers, optimizations, sorting, protocol management, and a number of other aspects of computers. Though the math involved from a calculation level is not complex (its mostly High school algebra) the theories and applications can be quite complex as a solid understanding of math through calculus will be of great benefit.

Can you get by without it, absolutely, and you shouldnt let a less then thorough knowledge of math hold you back, but if you had the chance, or the inclination I would study as much math as you could, calculus, numeric theory, linear algebra, combinatorics, practical applications, all of it has both practical and theoretical applications in a wide range of computer science.

I have known people who were highly successful on both sides of the fence (those without a strong focus on math, and those who went to school for physics or math), but in both groups they enjoyed numerical problems and learning about algorithms and math theory.

Share

edited Sep 16, 2010 at 16:38

answered Mar 2, 2010 at 23:45

GrayWizardx

21.1k ● 2 ● 33 ● 44

I second that. Though one can get by without it - will one regret his/her decision? If you expect to have even a moderately exciting/challenging job, those math skills you acquired will certainly help you. – Andras Vass Mar 3, 2010 at 0:14

Fair enough! GrayWizardx and others: As a self-learner, what do you recommend in terms of reading materials? I was massively turned off by the murderously bad way in which math was taught to me when I was younger. I don't want to lay blame elsewhere, but math was presented viciously. Now it's time to redeem. How EASY is it to learn the math outlined here (calculus, linear algebra, Discrete Math, etc.) on your own? Is there a particular set of textbooks recommended for this approach? A college degree is not an option anymore! – Emmel Mar 3, 2010 at 4:52

1 @Emmel, I actually saw "The Manga guide to Math" and "The Manga guide to Calculus" at a Barnes and Noble out there, and I know I have seen a "Complete Idiots Guide to Linear Algebra" the other day. As for actual resources, there use to be some great math websites (for students, etc) but I am not sure how many of them are still around. One thing I do every once in a while is go to a community college and "audit" the math classes (take it for pass/fail) and just get a refresher. Havent done it in 5-6 years, but it was pretty helpful the last time. – GrayWizardx Mar 3, 2010 at 7:28

26 votes

I have a maths degree, but I can't remember requiring that maths a single time in my career. It was useful in terms of training my mind for logical thinking, but I've not written any code using fluid dynamics, quantum theory or Markov Chains. (The last is the most likely to come up, I suspect.)

Most line-of-business developers won't need advanced maths most of the time. Sometimes knowing trigonometry can help, and certainly being able to understand enough maths to implement algorithms described mathematically can be important - but beyond that? Nah.

Don't forget that most programmers aren't advancing computer science - they're building applications. I don't need to know advanced engineering to drive a modern car, even though that car has almost certainly been improved *through* advanced engineering.

Share

answered Oct 1, 2008 at 12:40

community wiki
Jon Skeet

I agree with most of what you said, however, couldn't it be said that many aspects of programming can be markedly easier by understanding common math principles? Sometimes a formula for something is far easier to program than the algorithm to reach that same value...as long as you know that formula – BenAlabaster Dec 23, 2008 at 15:07

@balabaster: Sometimes - but I'd say that doesn't happen very often. Equally, you don't really need to know a formula beforehand to use it. Being *competent* at maths is good, so you can understand formulae when you see them, but you rarely need degree-level maths itself IMO. – Jon Skeet Dec 23, 2008 at 19:19

11 Math degree? Then you know it's not just numbers. I have two math degrees, and use the content often: boolean algebra to analyze/simplify logic, automata/language theory for DSL construction, algebra for big-O performance analysis, etc. – joel.neely Jan 13, 2009 at 12:52

14 votes

I would argue that having advanced logic (discrete) math can really help. That along with set theory. When dealing with common computer programs, these disciplines can help a lot. However, a lot of the other math I took in university was calculus, which as far as I can see, had very limited usage. Since 90% (or something like that) of programming is doing business apps with very simple math, I would say that for the most part, you can get by with very little math knowledge. However, a good understanding of boolean algebra, logic, discrete math, and set theory can really put you up to that next level.

Share

answered Oct 1, 2008 at 12:41

community wiki
Kibbee

14
votes

I'll go against the grain here and say "Yes"

I switch from Civil Engineering to programming (Concrete Sucks!). My math background consists of the usual first year stuff, second and third year Calculus(Diff EQ, volume integrations, Series, Fourier and Laplace transforms) and a Numerical Analysis course.

I find that my math is incredibly lacking for computer programming. There are entire areas of Discrete math and logic that I am missing, and I only survive due to an extensive library of textbooks, Wikipedia and Wolfram. Most advanced algorithms are based on advanced math, and I am unable to develop advanced algorithms without doing extensive research (Essentially the equivalent to a half-course worth of work.) I am certainly unable to come up with NEW algorithms, as I just don't have the mathematical foundations as the shoulders of giants upon which to stand.

Share

answered Oct 1, 2008 at 13:20

community wiki
Chris Cudmore

How many fundamental algorithms does an average programmer really write? I'd guess 80% of the programming jobs involve no math beyond basic algebra. – dbkk Oct 1, 2008 at 18:53

@dbkk: An average programmer's most common task is the design of the object model, the "business logic" of some domain. And such process is nothing else than mathematics, though it is not commonly reflected by those who perform the task. – Boris Stitnicky Oct 26, 2013 at 2:30

14
votes

It depends on what you are doing. If you do a lot of 3D programming, knowledge of 3D geometry is certainly necessary, don't you agree? ;-) If you want to create a new image format like JPG or a new audio format like MP3, you are also pretty lost if you can't understand a cosine or fourier transformation, as these are the basics most lossy compression are based on. Many other problems can be resolved better if you know your math rather well.

There are also many other programming tasks you will find do not need much math.

Share

edited Feb 16, 2012 at 19:57

community wiki
2 revs, 2 users 67%
Mecki

13

votes

If you find the subject fascinating enough to post this, just go ahead and start learning. The rest will come naturally.

Share

answered Mar 2, 2010 at 23:33

Daniel Situnayake
2,904 ● 2 ● 31 ● 38

9

votes

Yeah, there is no need for advanced mathematics - if you are programming commercial - off the shelf software.

However when dealing with hardcore stuff such as:

- Calculating trajectories to control a robot
- Creating AI-like applications to support uncertainty and automatic reasoning
- Playing with 3-D motion and graphics

Some advanced mathematics knowledge might come in handy. And it's not like they are "out-of-this world" problems.

I had to create a software to try to "predict" the necessary amount of paper for an office (and it was hell just to find out the best way to approximate values).

You have to be careful, though, because it is easy to get lost when using advanced things - there is a friend of mine who resorted to using Turing to store the state of a dynamic menu just to display it correctly - humm... perhaps he wnet too far in his imagination.

Share

answered Oct 1, 2008 at 12:48

community wiki
rshimoda

you could add things like compressing files or editing MP3 files or anything that requires actually understanding what the computer is doing, rather than just throwing a website together. – tloach Oct 1, 2008 at 13:35

4 How did he use Turing to do this? A ouija board or seance? – Jared Updike Mar 1, 2011 at 20:00

8 votes

What type of programming?

In my commercial experience, I have needed no advanced mathematics, but this is heavily dependent on the field you are in.

Computer graphics require a large amount of advanced mathematics. A lot of academic computer programming requires advanced mathematics.

So saying there tends to be a correlation between people who are good at mathematics and people who are good at programming.

I hope this wishy-washy answer helps.

Share

edited May 28, 2012 at 13:41

community wiki
2 revs, 2 users 70%
David Waters

6
votes

Mathematics are needed for developers in some fields but are almost useless in others.

If you are a game developer and have to work with physics a lot - understanding of math is crucial. If you are working with advanced visual controls - you could not do much without geometry. If you're planning to do some financial calculations - it would REALLY help to have solid knowledge of statistics.

On the other hand over last 5 years I had only 2 or 3 projects where ANY amount of math was required at all. Of these there was only 1 occurrence when a Google search did not help.

At the end of the day even financial calculations are very often something your clients do for you and give you formulas to implement.

So if you're in 'applied software' business you are likely to never use your math degree. If you're in academic software maths are crucial.

Share

answered Oct 1, 2008 at 12:42

community wiki
Ilya Kochetov

To "game developer" I'd add "simulation developer". I remember back in '93 driving all over town trying to find a math book sufficiently advanced to explain how Quaternions work to me, so that I could properly code a dead recokining algorithm. Thank God for the Internet today! – T.E.D. Dec 23, 2008 at 14:23

6
votes

I agree with Chris. I would say "Yes", also. But this depends on your market as stated above. If you are simply creating some basic "off-the-shelf" applications or writing tools to help your everyday work...then math isn't nearly as important.

Engineering custom software solutions requires lots of problem solving and critical thinking. Skills that are most definitely enhanced when a mathematics background is present. I minored in Math with my Computer Engineering degree and I give credit to all of my math-oriented background as to why I'm where I am today.

That's my 2 cents, I can tell from reading above that many would not agree. I encourage all to consider that I'm not saying you can't have those skills without a math background, I'm simply stating that the skills are side-effects of having such a background and can impact software positively.

Share

answered Oct 1, 2008 at 14:05

community wiki
MaTT

6
votes

In my experience math is required in programming, you can't get away from it. The whole of programming is based on math.

The issue is not black and white, but more colorful. The question is not whether or not you need math, but how much. The higher levels of math will give you more tools and open up your mind to different paths of though.

For example, you can program if you only known addition and subtraction. When multiplication is required, you will have to perform a lot of additions. Multiplication simplifies repetitive additions. Algebra allows one to simplify math before implementing it into programs. Linear Algebra provides tools for transforming images. Boolean Algebra provides mechanics for reducing all those `if` statements.

And don't forget the sibling to mathematics, Logic and Philosophy. Logic will help you make efficient use of `case` or `switch` statements. Philosophy will help you understand the thinking of the guy who wrote that code you are modifying.

Yes, you don't need much math to write programs. Some programs may require more math than others. More math knowledge will give you an advantage over those who have lesser understanding. In these times, people need every advantage they can get to obtain those jobs.

Share

answered Mar 3, 2010 at 0:31
Thomas Matthews
57.6k ● 17 ● 105 ● 163

6
votes

I've been programming for 8 years professionally, and since I was 12 as a hobby.

Math is not necessary, logic is. Math is horribly helpful though, to say it's not necessary is like saying that to kill a man, a gun isn't necessary, you can use a knife. Well, it is true, but that gun makes it a lot easier.

There are a couple bare minimums, which you should already meet. You need to know basic algebraic expressions and notation, and the common computer equivalents. For example, you need to know what an exponential is (3 to the 3rd is 27), and the common computer expression is 3^3. The common notations for algebra does change between languages, but many of them use a somewhat unified methodology. Others (looking at you LISP) don't. You also need to know order of operations.

You need to understand algorithmic thought. First this, then this, produces this which is used in this calculation. Chances are you understand this or you don't, and it's a fairly hard hurdle to jump if you don't understand it; I've found that this is something you 'get', and not really something you can learn. Conversely, some people don't 'get' art. They should not become painters. Also, there have been students in CS curriculum who cannot figure out why this does not work: x = z + w; z = 3; y = 5; It's not that they don't understand addition, it's that they aren't grasping the requirement of unambiguous express. If they understand it, the computer should too, right? If you can't see what's wrong with the above three lines, then don't become a programmer.

Lastly, you need to know whatever math is under your domain of programming. Accounting software could stop at basic algebra. If you are programming physics, you'll need to know physics (loosely) and math in 3-dimensional geometry (Euclidean). If you're programming architecture software, you'll need to know trigonometry.

This goes farther then math though; whatever domain you are programming for, you need to soundly understand the basics. If you are programming language analysis software, you'll need to know probability, statistics, grammar theory (multiple languages), etc.

Often times, certain domains need, or can benefit from, knowledge you'd think is unrelated. For example, if you were programming audio software, you actually need to know trigonometry to deal with waveforms.

Magnitude changes things also. If you are sorting a financial data set of 1000 items, it's no big thing. If it was 10 million records, however, you would benefit greatly from knowing vector math actually, and having a deep understanding of sorting at the binary level (how does a system sort alphabetically? How does it know 'a' is less than 'b'?)

You are going to find that as a programmer, your general knowledge base is going to explode, because each project will necessitate more learning outside of the direct sphere of programming. If you are squeamish or lazy about self-learning, and do not like the idea of spending 10+ hours a week doing essentially 'homework', do not become a programmer.

If you like thought exercises, if you like learning, if you can think about abstract things like math without a calculator or design without a sketchpad, if you have broad tastes in life and hobbies, if you are self-critical and can throw away 'favorited' ideas, if you like perfecting things, then become a programmer. Do not base this decision on math, but rather, the ability to think logically and learn. Those are what is important; math is just the by-product.

Share

answered Feb 25, 2011 at 18:23

community wiki
user634669

5
votes

Of course it depends on what kind of programmer you want to be, or better what kind of programmer your employers want you to be. I think calculus and algebra are essentials, statistic and linear programming is indeed a good tool to have in your briefcase, maybe analysis (derivative, integrals, functions...) could be done without. But if you want to know how things work skin-deep (electronics, for example, or some non trivial algorhytms) "advanced" math is something you'd better not go without anywhere.

Share

answered Oct 1, 2008 at 12:40

community wiki
Manrico Corazzi

5
votes

Most of the programming I have done involved physics simulations for research including things like electromagnetism, quantum mechanics and structural mechanics. Since the problem domains have advanced mathematics associated with them I would be hard pressed to solve them without using advanced mathematics.

So the answer to your question is - it depends on what you are trying to do.

Share

answered Oct 1, 2008 at 12:41

community wiki
WalkingRandomly

5
votes

Advanced maths knowledge is vital if you're going to be writing a new programming language. Or you need write your own algorithms.

However, for most day-to-day programming - from websites to insurance processing applications - only basic maths are **necessary**.

Share

answered Oct 1, 2008 at 12:42

community wiki
Oak

5
votes

Someone with a solid mathematical (which is not merely arithmetic) or logic background will cope well with algorithms, variable use, conditional reasoning and data structures.

- Not everyone can design a UI.
- Not everyone can make efficient code.
- Not everyone can comment and document clearly.
- Not everyone can do a good algorithm

Mathematics will help you to a point, but only to a point.

Share

answered Oct 1, 2008 at 12:57

community wiki
Unsliced

5

votes

I dont think advanced mathemetics knowledge is a requirement for a good programmer, but based on personal experience I think that programmers who have a better grasp at advanced maths also make better programmers. This may simply be due to a more logical mind, or a more logical outlook due to their experiences of solving mathematical problems.

Share

answered Oct 1, 2008 at 14:44

community wiki
Jimoc

5

votes

The fundamental concept of maths is the following, devising, understanding, implementation, and use of algorithms. If you cannot do maths then it is because you cannot do these things, and if you cannot do these things then you cannot be an effective programmer.

Common programming tasks might not need any specific mathematical knowledge (e.g. you probably won't need vector algebra and calculus unless you're doing tasks like 3D graphics or physics simulations, for example), but the underlying skillsets are identical, and lack of ability in one domain will be matched by a corresponding lack of ability in the other domain.

Share

answered Oct 1, 2008 at 17:38

community wiki
DrPizza

5

votes

Math is a toolbox for creating programs. I recommend Cormen's Introduction to Algorithms. It touches on the more "mathy" stuff.

```
- Greatest lowest limit (managing resources)
- Random variables (game programming)
- Topological sort (adjusting spreadsheets)
- Matrix operations (3d graphics)
- Number theory (encryption)
- Fast fourier transforms (networks)
```

Share

answered Jan 1, 2011 at 19:06

community wiki
user559910

4

votes

I don't think that higher math is a requirement for being a good programmer - as always it depends on what you are coding.

Of course if you are in 3D graphics programming, you'll need matrices and stuff. As author of business software, you'll probably need statistics math.

But being a professional programmer for almost 10 years (and another 10 years amateur) "higher math" is not something that I needed regularily. In about 99.8% of all cases it's just plus, minus, division and multiplication in some intelligent combinations - in most cases it's about algorithms, not math.

Share

answered Oct 1, 2008 at 12:43

community wiki
BlaM

4
votes

Learning higher math, for most programmers, is important simply because it bends your brain to think logically, in a step-by-step manner to get from one thing to another.

Very few programming jobs, though, require anything above high school math. I've used linear algebra once. I've never used calculus. I use algebra every day.

Share

answered Oct 1, 2008 at 12:44

community wiki
Lucas Oman

4
votes

Mathematical knowledge is often useful to a programmer, as are graphic design skill, puzzle-solving ability, work ethic and a host of other skills and traits. Very few programmers are good at everything that a programmer can possibly be good at. I wouldn't agree with any statement of the form "you're not a real programmer unless you can {*insert favorite programming ability here*}".

But I would be wary of a programmer who couldn't do Math. More so than of one who couldn't draw.

Share

answered Oct 1, 2008 at 13:14

community wiki
MusiGenesis

4
votes

I think it really depends on what you're trying to do, but IMHO, the CS and OS theory are more important than math here, and you really need only the math that they involve.

For example, there's a lot of CS background of scheduling theory and optimization that stands behind many schedulers in modern OSs. That is an example of something that would require some math, though not something super complicated.

But honestly, for most stuff, you don't need math. What you need is to learn the ability to think in base 2 and 16, such as the ability to mentally OR/AND. For example, if you have a byte and within that byte there are two 3-bit fields and 2 wasted bits, knowing which bits are in which fields are active when the byte value is something like 11 will make things slightly faster than having to use pen and paper.

Share

answered Mar 2, 2010 at 23:29

Uri

89.6k ● 51 ● 226 ● 322

Have you ever studied queueing theory? Have you ever studied memory managers and the statistical models that make developers choose one over the other? I don't necessarily disagree with the brunt of your answer, but your over-simplifying. – user124493 Mar 3, 2010 at 0:37

@San: I'm sure that I am oversimplifying, but I think that most systems developers never need to know these things at this level. I studied it at the basic level in my first two degrees, and somewhat deeper during my Ph.D., but it was never my forte, I prefer to leave it to the experts. The OS folks I knew had math skills that blew me away. – Uri Mar 3, 2010 at 1:15

which I think is very valid. i think there is a need to differentiate systems programming from OS programming. Tightly woven concepts, but different nonetheless. – user124493 Mar 3, 2010 at 1:50

4 votes

I began programming about the same time I entered my pre-algebra class.. So I wouldn't say math is all that important, though it can help in certain types of programming, especially functional.

I haven't taken Discrete Math yet, but I see a lot of theory stuff with programming written in a math-notation that is taught in this class.

Also, make sure you know how to calculate anything in any base, especially base 2, 8, and 16.

Also, one class that really brought home some concepts for me was this pre-programming class. We got taught unions, intersections, and all that happy stuff and It almost exactly parallels bitwise math. And we covered boolean logic very heavily. What I considered the most useful was when we learned how to reduce complex boolean statements. This was very handy:

```
(x|y) & (x|z) & (x|foo)
```

can be simplified to

```
x | (y & z & foo)
```

Which I previously did not quite grasp.

Share

answered Mar 2, 2010 at 23:34

Earlz

63.7k ● 100 ● 311 ● 506

4

votes

Well you generated a number of responses, and no I did not read them all. I am in the middle on this, no you certainly do not need math in order to be a programmer. Assembler vs device drivers in linux are no more or less complicated than the other and neither require math.

In no way shape or form do you need to take or pass a math class for any of this.

I will agree that the problem solving mindset for programming is quite similar to that of math solutions, and as a result math probably comes easily. or the contrary if math is hard then programming may be hard. A class or a degree or any pieces of paper or trophies are not required, going off and learning stuff, sure.

Now if you cannot convert from hex to binary to decimal quickly either in your head, on paper, or using a calculator you are going to struggle. If you want to get into networking and other things that involve timing, which kernel drivers often do but dont have to. You are going to struggle. I know of a very long list of people with math degrees and/or computer science, and/or engineering degrees that struggle with the rate calculations, bits per second, bytes per second, how much memory you need to do something, etc. To some extent it may be considered some sort of knack that some have and others have to work toward.

My bottom line is I believe in will power, if you want to learn this stuff you can and will, it is as simple as that. You dont need to take a class or spend a lot of money, linux and qemu for example can keep you busy for quite some time, different asm langauges, etc. crashable environments for kernel development, embedded, etc. You are not limited to that, but I dont believe that you have to run off and take any classes if you dont want to. If you want to then sure take some ee classes, some cs classes and some math classes..

Share

answered Mar 3, 2010 at 3:11

old_timer

71.4k ● 9 ● 98 ● 174

This isnt flame bait, but reality. Everything you need to know about digital electronics and software engineering from a programming perspective can be summed up with. 0, 1, and, or, not. those five things are what digital electronics and all programs operate on. You can build anything with a big box of fast nand gates. – old_timer Mar 3, 2010 at 3:14

4

votes

You need math. Programming is nothing more than math. Any findings of theoretical physics does not become a practical(applicable) implication, unless they are explained in terms of mathematical solutions. None of those can be solved computationally if they cannot be interpreted on computers, and more specifically on programming languages. Different languages are thus designed to solve specific problems. But for the general purpose and wide spread programming languages like java, c, c++ much of our programming tasks involve repetitive(continuous) solution to same problems like extracting values from database, text files, putting them on windows(desktop, web), manipulating same values, sometimes accessing some data from similar devices(but given different brand names, different port and a headache) etc which does not involve more than unitary method, and algebra(counter, some logic), geometry(graphics) etc. So it depends what you are trying to solve.

Share

answered Jan 3, 2011 at 23:18

community wiki
Pramod

3
votes

IMO, you probably need an aptitude for mathematics, without necessarily having much knowledge in the field. So the things you require to be good at maths are similar to the things you require to be good at programming.

But in general, I can't remember the last time I used any sort of advanced maths in day-to-day programming, so no.

Share

answered Oct 1, 2008 at 12:43

community wiki
Lehane

1 2 3 Next