# What are the core mathematical concepts a good developer should know? [closed]

Asked 16 years, 3 months ago  Modified 11 years, 6 months ago

Viewed 11k times

62

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.

Since Graduating from a very small school in 2006 with a badly shaped & outdated program (I'm a foreigner & didn't know any better school at the time) I've come to realize that I missed a lot of basic concepts from a mathematical & software perspective that are mostly the foundations of other higher concepts.

I.e. I tried to listen/watch the open courseware from MIT on Introduction to Algorithms but quickly realized I was missing several mathematical concepts to better understand the course.

So what are the core mathematical concepts a good software engineer should know? And what are the possible books/sites you will recommend me?

algorithm math language-agnostic

Share

edited Jun 12, 2013 at 12:48

Oded

**498k** ● 102 ● 893 ● 1k

asked Sep 9, 2008 at 15:33

Jose B.

**907** ● 1 ● 10 ● 13

Comments disabled on deleted / locked posts / reviews

## 27 Answers

Sorted by: Highest score (default)

62

votes

Math for Programmers. A good read.

Share

answered Sep 9, 2008 at 15:55

Gulzar Nazim

**52.2k** ● 26 ● 130 ● 170

Yegge's article is very thoroughly & it provides several other resources for further reading. – Jose B. Sep 10, 2008 at 15:35

4 by far, the best(est) article I read on this topic, answers all your confusion and dilemma about learning math. I wish could give this more than + Thank you so much Gulzar, please share some more articles like this if you have.
– Ramadheer Singh Dec 2, 2009 at 21:07

---

19 votes

Boolean algebra is fundamental to understanding control structures and refactoring. For example, I've seen many bugs caused by programmers who didn't know (or couldn't use) deMorgan's law. As another example, how many programmers immediately recognize that

```
if (condition-1) {
    if (condition-2) {
        action-1
    } else {
        action-2
} else {
    action-2
}
```

can be rewritten as

```
if (condition-1 and condition-2) {
    action-1
} else {
    action-2
}
```

Discrete mathematics and combinatorics are tremendously helpful in understanding the performance of various algorithms and data structures.

As mentioned by Baltimark, mathematical induction is very useful in reasoning about loops and recursion.

Set theory is the basis of relational databases and SQL.

By way of analogy, let me point out that carpenters routinely use a variety of rule-of-thumb techniques in constructing things like roofs and stairs. However, a knowledge of geometry allows you to solve problems for which you don't have a "canned" rule of thumb. It's like learning to read via phonetics versus sight-recognition of a basic vocabulary. 90+% of the time there's not much difference. But when you run into an unfamiliar situation, it's VERY nice to have the tools to work out the solution yourself.

Finally, the rigor/precision required by mathematics is very useful preparation for programming, regardless of specific technique. Again, many of the bugs in programming (or even specifications) that I've seen in my career have sloppy thinking at their root cause.

Share

answered Sep 9, 2008 at 16:57

joel.neely

**30.9k** ● 9 ● 57 ● 64

---

12 I would go with the fields that Landon stated:

votes

Discrete Math, Linear Algebra, Combinatorics, Probability and Statistics, Graph Theory

and add mathematical logic.

This would give you a grip on most fields of CS. If you want to go into special fields, you have to dive into some areas especially:

```
Computer graphics -> Linear Algebra
Gaming -> Linear Algebra, Physics
Computer Linguistics -> Statistics, Graph Theory
AI -> Statistics, Stochastics, Logic, Graph Theory
```

Share

answered Sep 9, 2008 at 17:13

Ralph M. Rickenbach

**13.1k** ● 5 ● 30 ● 49

---

10

votes

In order of importance:

- Counting (needed for loops)
- Addition, subtraction, multiplication, division.
- Algebra (only really required to understand the use of variables).
- Boolean algebra, boolean logic and binary.
- Exponents and logarithms (i.e. understand O(n) notation).

Anything more advanced than that is usually algorithm-specific or domain-specific. Depending on which areas you are interested in, the following may also be relevant:

- Linear algebra and trigonometry (3D visualization)
- Discrete mathematics and set theory (database design, algorithm design, compiler design).
- Statistics (well, for statistical and/or scientific/economic applications. possibly also useful for algorithm design).
- Physics (for simulations).

Understanding functions is also useful (don't remember what the mathematical term is for that area), but if you know how to program you probably already do.

My point being: A ten year old should know enough mathematics to be able to understand programming. There isn't really much math required for the basic understanding of things. It's all about the logic, really.

Share

answered Sep 9, 2008 at 18:34

Anders Sandvig

**21k** ● 16 ● 61 ● 74

---

9

votes

"Proof by induction" is a core mathematical concept for programmers to know.

Share

answered Sep 9, 2008 at 15:42

Baltimark

**9,272** ● 12 ● 38 ● 35

2 Really? For what programming-specific tasks is this required? – Anders Sandvig Sep 9, 2008 at 18:21

@Anders - Proving proofs in the case of algorithm development. – rjzii Sep 9, 2008 at 19:56

1 Luckily, induction proofs are some of the easiest imo. – mmcdole Nov 13, 2008 at 7:26

9 votes

Big O notation in general algorithm analysis, and in relation to standard collections (sorting, retrieval insertion and deletion)

Share

answered Sep 9, 2008 at 15:57

Corin Blaikie

18.1k ● 10 ● 39 ● 39

8 votes

For discrete math, here is an awesome set of 20 lectures from Arsdigita University. Each is about an hour and twenty minutes long.

Share

answered Sep 9, 2008 at 17:12

Corey

14.3k ● 7 ● 39 ● 35

7

votes

Start with what we CS folks call "discrete math". Calculus and linear algebra can come in quite handy too because they get your foot in the door to a lot of application domains. Once you've mastered those three, go for probability theory. Those 4 will get you to competency in 95% (I made that up) of application domains.

Share

answered Sep 9, 2008 at 15:37

Ben Collins

**20.7k** ● 18 ● 128 ● 190

Some people call "discrete math" "finite math". I believe they are both the same thing. – Thomas Owens Oct 24, 2008 at 15:31

7

votes

Concrete Mathematics covers most of the major topics. A good book on Discrete Math, like Rosen's Discrete Mathematics and Its Applications, will fill in any gaps.

Share

edited Jan 20, 2010 at 14:47

answered Sep 9, 2008 at 19:52

Bill the Lizard

**405k** ● 211 ● 572 ● 889

@mamama: I'd never noticed that was left out of CM before. Thanks for pointing it out. – Bill the Lizard Jan 20, 2010 at 14:49

5

votes

I think it depends on your focus. A few years ago I purchased the set of Art of Computer Programming by Donald Knuth. After looking at the books I realized pretty much everything is calculus proofs. If you're interested in developing your own generic algorithms and proofs for them, then I recommend being able to understand the above books since its what you'd be dealing with in that world. On the other hand if you only want/need to use various sorting/searching/tree/etc... routines then big O notation at a minimum, boolean math, and general algebra will be fine. If you're dealing with 3D then geometry and trig as well.

I tend to be more on the using side than making proofs, and while I'd like to think I've done some clever things over the years I've never sat down and developed a new sorting routine. The best advice I can give is learn what you need for your field, but expose yourself to higher levels so you know it exists and how much more there is to learn, you won't get much growth otherwise.

Share

answered Sep 9, 2008 at 16:26

ddowns

**316** ● 1 ● 5 ● 8

---

4

votes

I would say boolean logic. AND, OR, XOR, NOT. I found as programmer we use this more often than the rest of math concepts.

Share

answered Sep 9, 2008 at 16:11

Jimmy Chandra

**6,580** ● 4 ● 28 ● 38

---

3

votes

Basic Algebra and Statistics are good starting points, and the foundation for a lot of other fields.

Share

answered Sep 9, 2008 at 15:40

JosephStyons

**58.6k** ● 64 ● 167 ● 237

---

2

votes

Here is a simple one that baffles me when I see developers that don't understand it:

- Order of Operations

Share

answered Sep 9, 2008 at 17:13

Daniel Auger

**12.6k** ● 5 ● 53 ● 73

Agreed. I used to tutor intro CS courses and it took me awhile to realize that some students were having problems grasping this. It's such a natural thing for most of us to process that it's difficult at first to believe that some don't. – Bill the Lizard Sep 9, 2008 at 19:45

Same here. I always used to kind of stall out when I realized they didn't have any idea what I was talking about.
– Electrons_Ahoy Oct 24, 2008 at 20:47

---

2 votes

Chapter 1 of "The Art of Computer Programming" aims to provide exactly this.

Share

answered Sep 18, 2008 at 15:39

Michael Dorfman

**4,100** ● 1 ● 25 ● 29

---

1 vote

There was a book that was recommended...the title was something like Concrete Mathematics. It was recommended in a few questions.

Share

answered Sep 9, 2008 at 15:38

Thomas Owens

**116k** ● 99 ● 317 ● 436

it assumed that you have your fundamentals clear, it is introductory but advanced one. what else can we expect from knuth. – Ramadheer Singh Jun 30, 2010 at 18:11

1 vote

Back in school, on of my instructors said for business applications, all you need to know know add, subtract, multiply, and divide. All other formulas the requester will know and inform you what is needed. Now realize that this is for financing reporting and application focused school. To this day, this has held true for me. I have never once needed to know more than that.

Share

answered Oct 24, 2008 at 20:36

Mike Wills

**21.2k** ● 29 ● 96 ● 152

1 vote

Check the book Foundations of Computer Science
This book is authored by: Al Aho and Jeff Ullman and the entire book is available online.

This is what the authors say in their Preface about the goal of this book:

"*Foundations of Computer Science* covers subjects that are often found split
between a discrete mathematics course and a sophomore-level sequence in computer
science in data structures. It has been our intention to select the mathematical

foundations with an eye toward what the computer user really needs, rather than
what a mathematician might choose."

Share

answered Dec 17, 2009 at 12:08

sateesh

**28.6k** ● 7 ● 37 ● 45

---

1

vote

a site for brushing up on Math:
http://www.khanacademy.org/

Share

answered Apr 29, 2010 at 13:34

functor

**1,403** ● 2 ● 13 ● 12

---

0

votes

My math background is really poor (Geologist by training), but I took a discrete math class in high school and I use the concepts every day as a programmer. It is probably the most valuable class I took in all of my education as it relates to my current profession.

Share

answered Sep 9, 2008 at 15:39

JasonS

**23.9k** ● 9 ● 42 ● 47

---

0

votes

Discrete Math
Linear Algebra
Combinatorics

Probability and Statistics
Graph Theory

Share

answered Sep 9, 2008 at 17:03

Landon
**15.6k** ● 12 ● 39 ● 30

---

0
votes

- Boolean Algebra
- Set Theory
- Discrete Math

Share

answered Sep 9, 2008 at 17:38

mrbradleyt
**2,354** ● 7 ● 33 ● 38

---

0
votes

Well, that depends on what you goal is. As someone said, Linear Algebra, Combinatorics, Probability and Statistics and Graph Theory are important if you're into solving hard problems. Asymptotic growth of functions (bit-Oh notation) is very important. You will also need to master summations and series if you need to work on analyzing some more complex algorithms (see the appendix on Cormen&others Intro to Algorithms).

Even if you're into "Java for the enterprise" or "server-side PHP", you will find some Statistics and Algorithm Complexity (hence combinatorics, induction, summations, series, etc) useful when your boss wants you to get the

server to work faster, and adding new hardware doesn't seem to help. :-) I've been through that once.

Share

answered Oct 18, 2008 at 15:58



sergio_petralia

---

0

votes

- Boolean Algebra
- Set Theory

Share

answered Oct 24, 2008 at 15:29

Fortyrunner

**12.8k** ● 4 ● 34 ● 54

---

0

votes

Why is everybody including probability and statistics in the gold list without mentioning calculus? One cannot understand what probability and statistics are about without at least a working knowledge of limits, derivatives, integrals and series. And all in all, calculus (together with linear algebra) is the workhorse of *all* mathematics.

Share

edited Oct 24, 2008 at 19:47

answered Oct 24, 2008 at 15:58

Federico A. Ramponi

**47k** ● 30 ● 111 ● 133

0

votes

I think algorithms and theory are of great importance. Being able to come up with a fast, and *correct* solution is what differentiates good programmers from the rest. Also, being able to prove your algorithm (using standard proof techniques-- induction, contradiction, etc) is equally important.

Share

answered Nov 13, 2008 at 7:25

Scott Wegner

**7,473** ● 3 ● 41 ● 55

---

0

votes

Yeah, I would say a basic understanding of induction helps so that you understand what n represents in algorithms. Also some Logic and Discrete Structures is helpful.

Share

answered Apr 14, 2010 at 17:08

Aaron McAdam

**706** ● 2 ● 7 ● 20

0

votes

**Probability and Statistics** are very helpful if you ever have to do anything resembling machine learning.

I cover the basics in my "Computing Your Skill" blog post where I discuss how Xbox Live's TrueSkill ranking and matchmaking algorithm works.

Share

answered Apr 14, 2010 at 17:14

Jeff Moser

**20k** ● 6 ● 66 ● 85