Where can I learn about logarithms? [closed]

Asked 16 years, 3 months ago   Modified 8 years, 9 months ago

Viewed 5k times

20

Closed. This question is seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. It does not meet Stack Overflow guidelines. It is not currently accepting answers.

💡   We don't allow questions seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. You can edit the question so it can be answered with facts and citations.

Closed 8 years ago.

Improve this question

I hear logarithms mentioned quite a lot in the programming context. They seem to be the solution to many problems and yet I can't seem to find a real-world way of making use of them. I've read the Wikipedia entry and that, quite frankly, leaves me none the wiser.

So, where can I learn about the real-world programming problems that logarithms solve? Has

anyone got any examples of problems they faced that were solved by implementing a logarithm?

math   logarithm

Share

Improve this question

Follow

I'm voting to close this question as off-topic because it is not about programming. – High Performance Mark Mar 3, 2016 at 23:55

Can you explain why you think it isn't about programming? Note the sentence in bold in the second paragraph.
– Charles Roper Mar 4, 2016 at 1:15

19 Answers

Sorted by: Highest score (default) ⇕

▲

13

▼

🔖

Logarithms in programming are also frequently used in describing the efficiency of an algorithm using Big O notation.

For instance, a binary search algorithm would have a worst case scenario of O(log(n)) (on a sorted set), whereas a linear search's worst case is O(n)

answered Sep 23, 2008 at 15:52

Chris Marasti-Georg
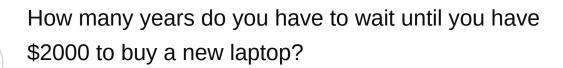**34.6k** ● 17 ● 93 ● 137

Yes of course, this is a great example. Any time you have something being split in half after each iteration, think Log! – Ben Hoffstein Sep 23, 2008 at 15:55

12

Say you've got $1000, and it's in a savings account with 2.4% interest.

How many years do you have to wait until you have $2000 to buy a new laptop?

$$1000 \times 1.024^X = 2000$$

$$1.024^X = 2$$

$$x = \log_{1.024} 2 = 29.23 \text{ years}$$

edited Feb 4, 2009 at 20:05

strager
89.9k ● 27 ● 138 ● 179

answered Sep 23, 2008 at 15:52

Mendelt
37.5k ● 6 ● 75 ● 97

1   I think ^ is clearer than ** to denote an exponent... to me at least – Chris Marasti-Georg Sep 23, 2008 at 15:57

This is very interesting and exactly what I meant by "real world". In trying to grok this answer I found the following very useful: math.about.com/od/formulas/a/compound.htm I ended up with the following to get back to 2000: 1000*(1 + 0.024)**29.2263362061568 – Charles Roper Sep 23, 2008 at 17:56

Fixed the typo's, thanks for the feedback. @Charles Roper you're getting it now. In other words log's tell you how many times you have to multiply a number by itself to get to another number. – Mendelt Sep 23, 2008 at 19:05

1   You're using inconsistent exponent operators now... :)
    – Charles Roper Sep 24, 2008 at 5:53

@Charles Roper: Damn, I shouldn't edit my posts late at night when drinking wine :-) – Mendelt Sep 24, 2008 at 7:43

In my own research I came upon a few useful resources:

10

Khan Academy logarithms section

This is a terrific set of lessons on logarithms. This comment from a 6th grader sums it up nicely:

> Thank you so much. This week, my math teacher told me to challenge myself, so I tried logarithms. At first I was like, 'I can't do this, it's too hard'. Then I watched the video, and now they're even fun! I'm in 6th grade, my math teacher is impressed. I can't thank you enough.

Ruby Quiz #105: Tournament Matchups

This article contains a good example of using a base 2 log to determine the number of rounds required to complete a knock-out tournament given *x* teams.

An Intuitive Guide To Exponential Functions & E

An excellent, intuitive (as you'd expect, given the title), guide to **e**, the base of the natural logarithm. Lots of illustrations and examples make this a gem of an article.

Demystifying the Natural Logarithm (ln)

This is the followup to the article to one about **e** and discusses the **natural logarithm (ln)** which, using the intuitive explanation given in the article, "gives you the time needed to reach a certain level of growth".

There's actually loads of good content on the **Better Explained** site. Truly, a splendid resource.

Another tool that I had actually come across before but since completely forgotten about is **Instacalc**. It seems to be by the same person - Kalid Azad - who authors the Better Explained site. It's a really useful tool when hacking about with maths.

answered Sep 24, 2008 at 17:52

Charles Roper
20.6k ● 20 ● 75 ● 105

+1 for the **Kalid Azad** mention! – Yatharth Agarwal Apr 24, 2012 at 14:08

5

Logs are a type of meta-arithmetic. It's a way of thinking of every number as a (possibly fixed) base raised to an exponent. Operations are performed on the exponents alone. This means that you can do multiplication and division by doing addition and subtraction of the logs. In other words, you put your data into log space, perform a suite of arithmetic, and pull it back into non-log space. If the floating point precision loss and the overhead of transforming in or out of log space is cheap, then you may have an overall win in time.

One slick trick you can do with logs is calculate out the number of characters a number will take when printed by taking the log-base-2 of the number and divide it by the log-base-10(2), which is constant time compared with a set of multiplies.

4

I've seen logarithms used for displaying tag clouds. This is a page that explains it:

[Tag Cloud Font Distribution Algorithm](#)

Share Improve this answer

Follow

4

I assume you have heard about logarithms with contexts to time consumation.

A concrete example would be search algorithms. Given a set of ordered data (think a sorted array of int's), you want to find the index key to a value in that data. We can benefit from the fact that the array is sorted (1, 2, 6, 192, 404, 9595, 50000 for example). Let's say we we want to find the index to the value 2. We can minimize our search space by culling (ignoring) half the array each step. We start this search by testing the value at the middle of the array. There are 7 values in the array, we then make the index 7/2 = 3.5 = 3 as int. array[3] is 192. The value we are looking for is 2, therefore we want to continue the search in the lower half of the search space. We

completely ignore index 4, 5, 6 since they are all higher than 192, and in turn also higher then 2. Now we have a search space that looks like (1, 2, 6). We then index into middle again (repeat process), and we find the 2 instantly. The search is complete, the index to 2 is 1.

This is a very small example, but it shows how such an algorithm works.

For 16 values, you need to search at maximum 4 times. For 32 values, you search max 5 times, 64 values 6 times and so on.. 1048576 values are searched in 20 steps. This is far quicker than having to compare each item in the array separately. Of course, this only works for sorted collections of data.

Share Improve this answer

Follow

answered Sep 23, 2008 at 16:05

Statement

4,038 ● 3 ● 40 ● 45

4

I recommend e: The Story of a Number for a good foundation of the importance of logarithms, their discovery and relevance to natural phenomena.

Share Improve this answer

Follow

answered Sep 23, 2008 at 16:16

Cade Roux

89.6k ● 40 ● 184 ● 266

Another way of looking at it is looking at the number of base multipliers in a number. I am sure you can see how this all relates in the following examples.

Decimal (base 10):

- $\log 10 \ (1) = 0$, $(10^0) = 1$
- $\log 10 \ (10) = 1$, $(10^1) = 10$
- $\log 10 \ (100) = 2$, $(10^2) = 100$
- $\log 10 \ (1000) = 3$, $(10^3) = 1000$
- $\log 10 \ (10000) = 4$, $(10^4) = 10000$
- $\log 10 \ (100000) = 5$, $(10^5) = 100000$

Binary (base 2):

- $\log 2 \ (1) = 0$, $(2^0) = 1$
- $\log 2 \ (2) = 1$, $(2^1) = 2$
- $\log 2 \ (4) = 2$, $(2^2) = 4$
- $\log 2 \ (8) = 3$, $(2^3) = 8$
- $\log 2 \ (16) = 4$, $(2^4) = 16$
- $\log 2 \ (32) = 5$, $(2^5) = 32$
- $\log 2 \ (64) = 6$, $(2^6) = 64$
- $\log 2 \ (128) = 7$, $(2^7) = 128$

Hexadecimal (base 16):

- $\log 16 \ (1) = 0$, $(16^0) = 1$

- log16 (16) = 1 , (16^1) = 16

- log16 (256) = 2 , (16^2) = 256

- log16 (4096) = 3 , (16^3) = 4096

- log16 (65536) = 4 , (16^4) = 65536

If you want to think in variables:

- log **N** (**X**) = **Y**

- (**N^Y**) = **X**

Share Improve this answer

Follow

answered Sep 23, 2008 at 16:18

Statement

4,038 ● 3 ● 40 ● 45

3

Many (many!) relationships in the real world are logarithmic. For instance, it would not surprise me if the distribution of reputation scores on Stack Overflow is log normal. The vast majority of users will have reputation scores of 1 and a handful of people will have unattainably high reputation. If you apply a logarithmic transformation to that distribution, it would likely be nearly a linear relation. A quick scan of https://stackoverflow.com/users?page=667 shows this to be true.
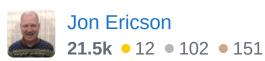
You might be more familiar with The Long Tail concept, which is an application of the logarithmic distribution.

Share Improve this answer

Follow

edited May 23, 2017 at 12:02

Community Bot

1 ● 1

The only problem I can recall is having to calculate the product of a column in SQL. SQL Server does not have a PRODUCT() aggregate function, so this was accomplished using a sum of the logarithms (using the LOG10() function) of each value. The main drawback was that all numbers in the column had to be positive and non-zero (you cannot calculate a logarithm on a negative number or zero).

2

Share   Improve this answer

Follow

Definitely a great transformation you mention here that can be expanded past just SQL. When products are really small you can often take the sum of the log. – nlucaroni Sep 23, 2008 at 15:51

The most obvious usage in every programming example is precision. Put simply, consider storing unsigned integers. How many bits do you need to store X? Well, the maximum value you can store in n bits is $2^n - 1$, so you can need $\log_2 X + 1$ bits to store X. Now you can pick short, int, word, long etc with ease.

2

answered Sep 23, 2008 at 15:53

Adam Wright

49.3k ● 12 ● 133 ● 152

2

One example, out of many : Calculating compound interests at a very small rate with a large number of periods.

You can do it the most straightforward way, even using fast exponentiation, but accuracy may suffer, due to the way floats are stored and calculating s * r power n still takes O(ln(n)) operations.

With logarithms, it's somewhat more accurate.

A = ln(s * r power n) = ln(s) + n * ln(r)
Two lookups in your logarithm database gives you ln(s) and ln(r), with ln(r) begin very small, and floats work at their best accuracy near 0 result = exp(A), a reverse lookup here.

It's also the only really efficient way if you work with non-integer exponents, to extract cubic roots for example.

answered Sep 23, 2008 at 16:08

community wiki
Johan Buret

Check out MIT's Open Courseware: [Introduction to Algorithms](#). Free educations. Awesome.

Share Improve this answer

Follow

answered Sep 23, 2008 at 16:11

Zach
24.8k ● 9 ● 45 ● 50

Looks very interesting indeed, but does it cover logarithms?
– [Charles Roper](#) Sep 23, 2008 at 16:22

One of the more "cool" applications of logarithms I've found is [Spiral Storage](#). It's a hash table that allows you to split one bucket at a time as the table grows, relocating less than half of the records in that bucket to the same, new bucket. Unlike linear hashing, where the performance varies cyclically and all of the buckets tend to be split at around the same time, spiral hashing allows nice, smooth growth of the table.
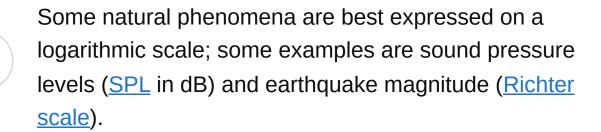
It was published about 30 years ago by G. N. N. Martin, about whom I haven't been able to learn much besides the fact that he also invented [Range Encoding](#). Seems like a smart guy! I haven't been able to get a copy of his original paper, but Per-Åke Larson's paper ["Dynamic hash tables"](#) has a very clear description.

Share Improve this answer

answered Sep 23, 2008 at 16:41

▲

1

▼

🔖

🕐

Logarithms are used quite often in charts and graphs, when one or both axes cover a large range of values.

Some natural phenomena are best expressed on a logarithmic scale; some examples are sound pressure levels (SPL in dB) and earthquake magnitude (Richter scale).

Share   Improve this answer

Follow

answered Sep 23, 2008 at 16:06
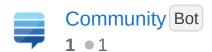
Mark Ransom

308k ● 44 ● 416 ● 647

▲

1

▼

🔖

🕐

As an example of what Chris is talking about, an algorithm that changes complexity based on the number of bits in a value is (probably) going to have an efficiency described by O(log(n)).

Another everyday example of exponents (and hence logarithms) is in the format of IEEE floating point numbers.

A logarithmic function is simply the inverse of an exponential function, in the same sense that subtraction is the inverse of addition. Just as this equation:

```
a = b + c
```

states the same fact as this equation:

```
a - c = b
```

this equation:

```
b ** p = x
```

(where `**` is raising to a power) states the same fact as this equation:

```
log [base b] (x) = p
```

Although `b` can be any number (e.g. `log [base 10] (10,000) = 4`) the "natural" base for Mathematics is `e`

(2.718281828...) about which [see here](#).

"Common" logarithms, used more in engineering, use a base of 10. A quick-and-dirty (emphasis on dirty) interpretation of the common (base 10) logarithm of some number x is that it is one less than the number of decimal digits required to express numbers the size of x.

Share Improve this answer

Follow

answered Sep 24, 2008 at 2:14

joel.neely

30.9k ● 9 ● 57 ● 64

Demystifying the Natural Logarithm (ln) at BetterExplained is the best i have found. It clears the concepts from the base and help you understand the underlying concepts. After that everything seems a cakewalk.

0

Share Improve this answer

Follow

answered Jan 17, 2009 at 13:12

Navneet

11.8k ● 4 ● 22 ● 22

Here are some sites that I have used:

0

- http://www.helpalgebra.com/articles/propertiesoflogarithms.htm

- http://www.math.unc.edu/Faculty/mccombs/web/alg/classnotes/logs/lognotation.html

- http://www.math.unc.edu/Faculty/mccombs/web/alg/classnotes/logs/logprops.html
- http://abacus.bates.edu/acad/acad_support/msw/exps_and_logs.pdf
- http://people.hofstra.edu/Stefan_Waner/Realworld/calctopic1/logs.html

I have used logarithms for calculating the yearly appreciation on a house to determine whether the seller was being fair.

House Appreciation Equations

Here is the basic equation:

- Previous price = p
- New price = n
- Appreciation rate = r
- Years of appreciation = y

```
p * (1 + r)^y  = n
```

So, if the price 6 years ago was $191,000 (by checking your couty auditor's site) and the asking price is $284,000, what is the appreciation rate (which would not take into account any one-time improvement costs)?

```
191,000 * (1 + r)^6 = 284,000
```

```
(1 + r)^6 = 284,000 / 191,000 = 1.486

Using a property of exponents and
logarithms…

6 ( log (1 + r) ) = log 1.486
log (1 + r) = (log 1.486) / 6 = 0.02866

Using another property of exponents and
logarithms…

10 0.02866 = 1 + r
1.068 = 1 + r
r = 1.068 – 1 = 0.068 = 6.8%  (kind of
high!)
```

To determine what a reasonable price would be…use 4% and allow for whatever improvements they made (which should be listed on the web id they were major…but it wouldn't include bathroom/kitchen remodeling, etc.)

```
191,000 * (1 + 0.04)^6 = n
n = 241,675 + reasonable cost of
improvement
which of course will depreciate over time
and should not represent 100% of the
cost of the improvement
```

Share Improve this answer

Follow

answered Jan 29, 2009 at 15:24

a belle