Maths for Computer Science
Logs

Denis TRYSTRAM

Logarithms: the scaling functions

Definition

b is the base (positive real number).
$log(x)$ is defined as the inverse of the exponentiation $f(x) = b^x$:
$x = b^{log_b(x)}$

Using this definition and the basic property of the exponential, we can establish most existing properties of the log.

Logarithms: the scaling functions

Definition

b is the base (positive real number).
$log(x)$ is defined as the inverse of the exponentiation $f(x) = b^x$:
$x = b^{log_b(x)}$

Using this definition and the basic property of the exponential, we can establish most existing properties of the log.

- $log_b(x \cdot y) = log_b(x) + log_b(y)$
- $log_b(1) = 0$ is a consequence of this definition, not by convention!
- $\log_a(x) = \log_a(b) \log_b(x)$

Going further

Another useful expression of $n^{\log_a(b)}$

- $n^{\log_a(b)} = b^{\log_a(n)}$

- Draw the shape of the *log* function
- What are the links between the sum of the harmonic series and the *log*?

Geometrical interpretation

- draw the integral of $1/x$
- give the interpretation of the multiplicative rule

Going further

- natural logarithm, base $e = 2.71828...$
- Base 2: the "natural" log of $2 = 0.69314...$
- base 10: $log(10^{-2}) = -2$

Most of them are irrational.
Example: prove that $log(2)$ is irrational, by contradiction:
Assume it is p/q, then $10^{p/q} = 2$, thus $10^p = 2^q$
impossible since the first ends by the digit 0 and the second by $2, 4, 6$ or 8
Using the same argument, $log(3)$ is irrational since the powers of 3 are odds.

Use of logs (in Computer Science)

- space needed for coding integers, base 2

Examples of use

We study here Maths concepts and results that can be useful for solving problems in algorithmic.

Principle of the divide and conquer paradigm (Analysis by the Master theorem)

Motivation

problems whose input can be decomposed

[1] Generally, the k sub-problems are solved by the same method (recursively)

Examples of use

We study here Maths concepts and results that can be useful for solving problems in algorithmic.

Principle of the divide and conquer paradigm (Analysis by the Master theorem)

Motivation

problems whose input can be decomposed

Principle for a problem of size n:

If n is "small" enough, we compute the value using any existing method. Otherwise

1. Decompose the problem into k sub-problems of size n_i.
2. Solve the k sub-problems of sizes n_i $(1 \leq i \leq k)$[1]
3. Rebuild the original solution form the k partial solutions.

[1] Generally, the k sub-problems are solved by the same method (recursively)