



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)  
[Donate to Wikipedia](#)  
[Wikipedia store](#)

Interaction  
[Help](#)  
[About Wikipedia](#)  
[Community portal](#)  
[Recent changes](#)  
[Contact page](#)

Tools  
[What links here](#)  
[Related changes](#)  
[Upload file](#)  
[Special pages](#)  
[Permanent link](#)  
[Page information](#)  
[Wikidata item](#)  
[Cite this page](#)

Print/export  
[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)


Languages  [Add links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#)

[More▼](#)



# Approximate counting algorithm

From Wikipedia, the free encyclopedia

The **approximate counting algorithm** allows the counting of a large number of events using a small amount of memory. Invented in 1977 by [Robert Morris \(cryptographer\)](#) of [Bell Labs](#), it uses [probabilistic techniques](#) to increment the [counter](#). It was fully analyzed in the early 1980s by [Philippe Flajolet](#) of [INRIA](#) Rocquencourt, who coined the name **Approximate Counting**, and strongly contributed to its recognition among the research community. The algorithm is considered one of the precursors of streaming algorithms, and the more general problem of determining the frequency moments of a data stream has been central to the field.

## Contents [\[hide\]](#)

- [1 Theory of operation](#)
- [2 Algorithm](#)
- [3 Applications](#)
- [4 Sources](#)

## Theory of operation [\[edit\]](#)

Using Morris' algorithm, the counter represents an "[order of magnitude](#) estimate" of the actual count. The approximation is mathematically [unbiased](#).

To increment the counter, a [pseudo-random](#) event is used, such that the incrementing is a probabilistic event. To save space, only the exponent is kept. For example, in base 2, the counter can estimate the count to be 1, 2, 4, 8, 16, 32, and all of the [powers of two](#). The memory requirement is simply to hold the [exponent](#).

As an example, to increment from 4 to 8, a pseudo-random number would be generated such that a probability of .25 generates a positive change in the counter. Otherwise, the counter remains at 4.

The table below illustrates some of the potential values of the counter:

Stored binary value of counter	Approximation	Range of possible values for the actual count
0	1	0, or initial value
1	2	1 or more
10	4	2 or more
11	8	3 or more
100	16	4 or more
101	32	5 or more

If the counter holds the value of 101, which equates to an exponent of 5 (the decimal equivalent of 101), then the estimated count is 2^5, or 32. There is a very low probability that the actual count of increment events was 5 (which would imply that an extremely rare event occurred with the pseudo-random number generator, the same probability as getting 10 consecutive heads in 10 coin flips). The actual count of increment events is likely to be around 32, but it could be infinitely high (with decreasing probabilities for actual counts above 32).

## Algorithm [\[edit\]](#)

When incrementing the counter, "flip a coin" the number of times of the counter's current value. If it comes up "Heads" each time, then increment the counter. Otherwise, do not increment it.



This can be done programmatically by generating "c" pseudo-random bits (where "c" is the current value of the counter), and using the logical AND function on all of those bits. The result is a zero if any of those pseudo-random bits are zero, and a one if they are all ones. Simply add the result to the counter. This procedure is be executed each time the request is made to increment the counter.

## Applications [\[edit\]](#)

The algorithm is useful in examining large data streams for patterns. This is particularly useful in applications of

[data compression](#), sight and sound recognition, and other [artificial intelligence](#) applications.

## Sources [\[edit\]](#)

- Morris, R. Counting large numbers of events in small registers. Communications of the ACM 21, 10 (1977), 840–842
- Flajolet, P. Approximate Counting: A Detailed Analysis. BIT 25, (1985), 113-134 [\[1\]](#) 
- Michael F., Chung-Kuei L., Prodinge, Approximate Counting via the Poisson-Laplace-Mellin Method [\[2\]](#) 

Categories: [Randomized algorithms](#)

This page was last modified on 17 March 2015, at 14:27.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

