Ruby trunk - Feature #8748

Integer#popcount (Fixnum#popcount and Bignum#popcount)

08/07/2013 09:51 PM - akr (Akira Tanaka)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

How about adding Integer#popcount method?
(actually Fixnum#popcount and Bignum#popcount)

0.popcount #=> 0
1.popcount #=> 1
255.popcount #=> 8
256.popcount #=> 1
(10*100).popcount #=> 105
(257*257).popcount #=> 999

It counts the number of one bits in the integer.
If the integer is negative, the one bits in the absolute number is counted.

popcount has various applications.
Hamming distance, rank/select for succinct data structure,
brightness of monochrome image, etc.

In general, popcount is useful when an array is encoded as an integer.

Several lower layers provides this feature.
gcc and clang has __builtin_popcount.
Intel and AMD provides popcnt instruction.

Several languages and libraries provides this feature:

absolute number: Mathmatica(DigitCount)
two's complement: Java(java.math.BigInteger#bitCount), Scala(bitCount), CommonLisp(logcount), CLN(logcount)
other behavior: GMP(mpz_popcount), Haskell(popCount), Scheme(bitwise-bit-count)
fixed size: gcc (__builtin_popcount), Intel/AMD(popcnt), Java(java.lang.Integer.bitCount)

For negative numbers, my implementation counts bits in abs(n).
I think this is easy to understand, at least.
However many software counts bits in two's complement representation.

There are several names.
I think popcount is popular but bitcount is also a possible name.
I don't like logcount.

Any comments?

Details of the other software:

Mathmatica has DigitCount which can be used as popcount.
n.popcount can be implemented as DigitCount[n, 2, 1].
It seems work for abs(n). (I tested with Wolfram Alpha.)
http://reference.wolfram.com/mathematica/ref/DigitCount.html

Java has bitCount method in java.lang.Integer and java.math.BigInteger.
java.lang.Integer counts one-bits in two's complement representation
(so it is not applicable for infinite precision integer).
java.math.BigInteger counts bits which is different to sign bit in
two's complement representation.
http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html#bitCount(int)
http://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html#bitCount()

Scala has bitCount method too.
It works as Java.
http://www.scala-lang.org/api/current/index.html#scala.math.BigInt

CommonLisp has logcount function.
http://www.lispworks.com/documentation/HyperSpec/Body/f_logcou.htm

CLN has logcount function.
http://www.ginac.de/CLN/cln.html#Exact-numbers

GMP has mpz_popcount.
It returns a some constant for negative values.
http://gmplib.org/manual/Integer-Logic-and-Bit-Fiddling.html#Integer-Logic-and-Bit-Fiddling

Haskell has popCount.
It seems hang for negative values.
http://www.haskell.org/ghc/docs/7.6.2/html/libraries/base/Data-Bits.html#t:Bits

Scheme has bitwise-bit-count.
It returns negative result for negative values.
http://www.r6rs.org/final/html/r6rs-lib/r6rs-lib-Z-H-12.html#node_sec_11.1

History

#1 - 08/31/2013 03:51 PM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I don't see the needs to add methods to use integers as bit-arrays.

Matz.

#2 - 08/31/2013 07:54 PM - Anonymous

The issue here seems to be, whether BitArray (like https://github.com/peterc/bitarray) is desirable in stdlib or core.

#3 - 08/31/2013 08:57 PM - alexeymuranov (Alexey Muranov)

boris_stitnicky (Boris Stitnicky) wrote:

> The issue here seems to be, whether BitArray (like https://github.com/peterc/bitarray) is desirable in stdlib or core.

+1 for something like BitArray in core (edit: or in stdlib).

-1 for using integers as bit arrays. (IMHO)

Files

popcount.patch	8.02 KB	08/07/2013	akr (Akira Tanaka)