Ruby - Bug #1243

1 is prime

03/04/2009 12:37 AM - yugui (Yuki Sonoda)

Status:	Closed	
Priority:	Normal	
Assignee:	yugui (Yuki Sonoda)	
Target version:	1.9.2	
ruby -v:	ruby 1.9.1p0 (2009-02-22 revision 22551) [i386-darwin9.6.0]	**Backport:**

Description

=begin
Prime.prime? always returns true for n < 2
=end

History

#1 - 03/04/2009 01:07 AM - yugui (Yuki Sonoda)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin
Applied in changeset r22741.
=end

#2 - 03/04/2009 01:30 AM - headius (Charles Nutter)

=begin
The number 1 is not considered prime:

http://en.wikipedia.org/wiki/Prime_number#Primality_of_one
=end

#3 - 03/04/2009 01:41 AM - daz (Dave B)

=begin

- value = -value if value < 0 <--- NOT required ?
- return false if value < 2

I think your negative guard is not required?

All negatives, 0 and 1 are non-prime, therefore all < 2 are non-prime.

http://en.wikipedia.org/wiki/Prime_number

daz

=end

#4 - 03/04/2009 05:17 PM - yugui (Yuki Sonoda)

=begin
On 3/4/09 1:40 AM, Dave B wrote:

> - value = -value if value < 0 <--- NOT required ?
> - return false if value < 2
>
> I think your negative guard is not required?

For a arbitrary positive prime number p, -p is a prime element in the
ring of integers. On the other hand, it is sure that just saying "prime
numbers" means positive ones.

(http://en.wikipedia.org/wiki/Prime_number)

> Without further specification, however, “prime number” always means a
> positive integer prime

I am at a loss whether -2.prime? should return true or false? Which is
more useful for Rubyists?

--
Yugui yugui@yugui.jp
http://yugui.jp

=end

#5 - 03/04/2009 05:18 PM - yugui (Yuki Sonoda)

=begin
On 3/4/09 1:29 AM, Charles Nutter wrote:

> The number 1 is not considered prime:

Sorry, I wanted to mean that the implementation of Prime regarded 1 as a
prime.

--
Yugui yugui@yugui.jp
http://yugui.jp

=end

#6 - 03/04/2009 05:59 PM - daz (Dave B)

=begin
Ring theory defines "prime element" as a special case.
Someone else might request adding a "prime_element?" or "ring_prime?" method, later.
Maybe they could use (-7.abs.prime?) for their specific application.

The ordinary definition of prime must return false for (-2.prime?)
A (true) result for (-2.prime?) or any other negative numbers is wrong,
regardless of how useful it would be to Rubyists. ;))

daz

=end

#7 - 03/06/2009 11:12 PM - RobertDober (Robert Dober)

=begin
On Wed, Mar 4, 2009 at 9:15 AM, Yugui (Yuki Sonoda) yugui@yugui.jp wrote:

> On 3/4/09 1:40 AM, Dave B wrote:
>
> > - value = -value if value < 0 <--- NOT required ?
> > - return false if value < 2
> >
> > I think your negative guard is not required?
>
> For a arbitrary positive prime number p, -p is a prime element in the
> ring of integers. On the other hand, it is sure that just saying "prime
> numbers" means positive ones.
>
> (http://en.wikipedia.org/wiki/Prime_number)
>
> > Without further specification, however, “prime number” always means a
> > positive integer prime
>
> I am at a loss whether -2.prime? should return true or false? Which is
> more useful for Rubyists?

IIRC 1 was discarded as prime because it messed up the uniqueness of
factorization. If we want to keep this spirit I see no solution for
negative numbers.

One could maybe define that all negative numbers are composed with the
exception of -1, which is prime ARRRGH and the additional rule that -1
can only occur ? times (in the regexp sense of ?) in a factorization.
But maybe it makes more sense to say x.prime? if and only if x.abs.prime?
and
for all x: x.primefactors.count{ | f| f.negative? } < 2

I guess these are lousy ideas, does anyone have something better to suggest?

Cheers
Robert

> --
> Yugui yugui@yugui.jp
> http://yugui.jp

--
There are some people who begin the Zoo at the beginning, called
WAYIN, and walk as quickly as they can past every cage until they get
to the one called WAYOUT, but the nicest people go straight to the
animal they love the most, and stay there. ~ A.A. Milne (from
Winnie-the-Pooh)

=end

#8 - 03/08/2009 02:38 AM - RobertDober (Robert Dober)

=begin
On Fri, Mar 6, 2009 at 8:42 PM, Jacob Fugal lukfugl@gmail.com wrote:

> I have a hard time imagining a situation where asking if a negative
> number is prime would be necessary. Code the cares about prime numbers
> almost by design implies that it expects positive integers. In that
> mindset, I call YAGNI.
>
> There is no compelling argument (to me) to have the Ruby standard
> library definition of prime be inconsistent with the mathematical
> definition. A prime number -- not a prime factor or a prime element of
> the ring of integers or a prime element of any other arbitrary ring --
> is a natural number (that is, a positive -- or non-negative, it
> doesn't matter here -- integer) which has exactly two distinct natural
> number divisors: 1 and itself. 1 is not prime and negative integers
> are not prime. Rationals or Floats not coinciding with an integer are
> not prime. That should be the definition of "prime?".
>
> In the rare case where someone does need to check if the absolute
> value of a negative integer x is prime, x.abs.prime? is not a horrible
> burden, and it reveals the intent that negative numbers are acceptable
> input to boot.
>
> Jacob Fugal

I tend to agree, I was taken away by the "challenge" to find a
meaningful definition for primes < 0. And when I failed to do so I
asked for help, stupid somehow, is it not, LOL.

Cheers
Robert

=end