

## Ruby trunk - Feature #10354

### Optimize Integer#prime?

10/10/2014 03:22 AM - marcandre (Marc-Andre Lafortune)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	yugui (Yuki Sonoda)	
<b>Target version:</b>		
<b>Description</b>		
<p>Nick Slocum shows in <a href="https://github.com/ruby/ruby/pull/736">https://github.com/ruby/ruby/pull/736</a> that Integer#prime? can be optimized quite a bit.</p> <p>First, that's because there are some basic things to avoid in the current lib, like needlessly capturing blocks and there's a useless loop do too.</p> <p>I'm attaching a patch that fixes many of these things.</p> <p>Even after these fixes applied, Nick's version is still faster and I don't see why we would not use it for Fixnum#prime?</p> <p>For Bignum#prime?, since division costs more, we can go slightly faster with the following implementation:</p> <pre>class Integer   # Returns true if +self+ is a prime number, else returns false.   def prime?     return true if self == 2     return false if self % 2 == 0    self % 3 == 0    self &lt; 2     skip_division = true     (5..(self**0.5).floor).step(2) do  i        return false if skip_division &amp;&amp; self % i == 0       skip_division = !skip_division     end     true   end end</pre>		
<b>Related issues:</b>		
Is duplicate of Ruby trunk - Feature #5378: Prime.each is slow		<b>Assigned</b>

#### Associated revisions

##### Revision 0eebb8f1 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48767 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 48767 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

##### Revision 48767 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

##### Revision 48767 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

##### Revision 48767 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

**Revision 48767 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

**Revision 48767 - 12/10/2014 08:38 PM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Remove useless loop and block capture. See [#10354]

**Revision 8744d516 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug #10354]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52200 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 52200 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug #10354]

**Revision 52200 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug #10354]

**Revision 52200 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug #10354]

**Revision 52200 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug #10354]

**Revision 52200 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)**

- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug #10354]

---

## History

**#1 - 10/10/2014 03:40 AM - marcandre (Marc-Andre Lafortune)**

Oops, it would help if I gave the right implementation for Bignum#prime? See below

Here are some benchmarks:

Benchmark over 1..1\_000\_000

Current prime lib: 15.54

Fixed prime lib: 7.13

Nick's: 2.49

Benchmark for ((1 << 65) + 5).prime?:

Current prime lib: 11.83

Fixed prime lib: 6.82

Nick's: 5.32

Mine: 4.91

```
class Integer
  # Returns true if +self+ is a prime number, else returns false.
  def prime?
    return true if self == 2
    return false if (self % 2 == 0) || (self % 3 == 0) || (self < 2)
    skip_division = 3
    (5..(self**0.5).floor).step(2) { |i|
      if (skip_division -= 1) == 0
        skip_division = 3
      next
    end
    return false if self % i == 0
  }
  true
end
```

end

## #2 - 10/12/2014 08:44 PM - Anonymous

I rewrote #prime? again using a better algorithm. It's about 2x faster. This version is for Integer, but could be reworked for BigNum.

```
class Integer
  def prime?
    return self >= 2 if self <= 3
    return false if self % 2 == 0 or self % 3 == 0
    (5..(self**0.5).floor).step(6).each do |i|
      if self % i == 0 || self % (i + 2) == 0
        return false
      end
    end
    true
  end
end
```

Benchmarked with ruby-2.1.3

my first version: 4.27

this latest version: 2.76

## #3 - 10/29/2014 03:02 AM - marcandre (Marc-Andre Lafortune)

Dear Yugui,

Do you have an objection that I commit my fixes (which are really bug fixes) while you decide if you want to go in the direction of Nick's optimization?

Thanks

## #4 - 02/05/2015 07:47 PM - marcandre (Marc-Andre Lafortune)

- Is duplicate of Feature #5378: Prime.each is slow added

## #5 - 10/20/2015 03:18 AM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Closed

Applied in changeset [r52200](#).

- 
- lib/prime.rb: Optimize Integer#prime? Patch by Nick Slocum [Bug [#10354](#)]

## Files

---

prime_opt.patch	1.55 KB	10/10/2014	marcandre (Marc-Andre Lafortune)
-----------------	---------	------------	----------------------------------