

## Ruby master - Bug #14699

### Subtle behaviors with endless range

04/19/2018 03:40 PM - mame (Yusuke Endoh)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b> 2.6	
<b>ruby -v:</b>	<b>Backport:</b> 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
<b>Description</b> Currently, some Range's methods behaves weirdly with endless range.  <b>Range#max</b> Range#max always returns nil. Is this okay, or is another result expected? <pre>p (0..).max #=&gt; nil p (0..).max(3) #=&gt; nil p (0..).max { a, b  a &lt;=&gt; b } #=&gt; nil</pre> Note that (0..).min { a, b  a <=> b } gets stuck. I think Range#min and Range#max should behave the same if a block are passed, but I'm uncertain what behavior is preferable.  <b>Range#last</b> Range#last returns nil if no argument is passed. But it gets stuck if a length argument is passed. Is this okay? <pre>p (0..).last #=&gt; nil p (0..).last(3) #=&gt; stuck</pre> <b>Range#size</b> Range#size returns nil for endless range. I think this is somewhat reasonable because ("a".."z").size returns nil.	
<b>Related issues:</b> Related to Ruby master - Feature #12912: An endless range `(1..)` <span style="float: right;"><b>Closed</b></span>	

#### Associated revisions

##### Revision 34261930 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)

range.c: Range#size now returns Float::INFINITY if it is endless

Fixes [Bug #14699]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63715 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 63715 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)

range.c: Range#size now returns Float::INFINITY if it is endless

Fixes [Bug #14699]

##### Revision 63715 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)

range.c: Range#size now returns Float::INFINITY if it is endless

Fixes [Bug #14699]

#### History

##### #1 - 04/19/2018 03:41 PM - mame (Yusuke Endoh)

- Related to Feature #12912: An endless range `(1..)` added

**#2 - 04/19/2018 05:24 PM - marcandre (Marc-Andre Lafortune)**

I believe `(0..).max(3)` and `(0..).max { |a, b| a <=> b }` can not return nil. They should either hang or (seems more useful) raise an error, same as `min`. I would have `(0..).last(3)` also raise an error.

`(0..).size` must return `Float::INFINITY`, not nil. The only reason why `("a".."z").size` returns nil was that I was too lazy to code a valid calculation that handled all the strange corner cases of the way we iterate on strings.

**#3 - 05/17/2018 05:21 AM - naruse (Yui NARUSE)**

- *Target version set to 2.6*

Need to be discussed with mrkn before December.

**#4 - 06/21/2018 06:13 AM - matz (Yukihiko Matsumoto)**

I agree with [marcandre \(Marc-Andre Lafortune\)](#) here. Let's raise exceptions.

Matz.

**#5 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)**

- *Status changed from Open to Closed*

Applied in changeset [trunk|r63715](#).

---

range.c: `Range#size` now returns `Float::INFINITY` if it is endless

Fixes [Bug [#14699](#)]