

## Ruby trunk - Bug #7352

### Array#bsearch test failure on Range (32bits MinGW)

11/15/2012 08:48 AM - luislavena (Luis Lavena)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> mame (Yusuke Endoh)	
<b>Target version:</b> 2.0.0	
<b>ruby -v:</b> ruby 2.0.0dev (2012-11-15 trunk 37656) [i386-mingw32]	<b>Backport:</b>
<b>Description</b> =begin Hello,  After <a href="#">r37655</a> , I noticed a failing test on RubyInstaller CI for x86:  test_bsearch_for_float(TestRange) [C:/Users/Luis/Code/ruby/ruby/test/ruby/test_range.rb:392]: Expected -1.7976931348623157e+308 to be >= NaN.  This do not fail on x64, both running with GCC 4.7.2  I don't have Linux test results to compare at this time, but wanted to raise awareness of this issue.  Thank you =end	
<b>Related issues:</b> Related to Ruby trunk - Feature #4766: Range#bsearch <span style="float: right;">Closed 05/23/2011</span>	

#### Associated revisions

##### Revision 412b0237 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37662 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 37662 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

##### Revision 37662 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

**Revision 37662 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)**

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

**Revision 37662 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)**

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

**Revision 37662 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)**

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

**Revision 37662 - 11/15/2012 01:50 PM - mame (Yusuke Endoh)**

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

## History

---

### #1 - 11/15/2012 12:30 PM - mame (Yusuke Endoh)

Thank you Luis, I'll try it on windows.

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

### #2 - 11/15/2012 04:40 PM - phasis68 (Heesob Park)

I guess this bug is related with the optimization of GCC 4.7.2.  
The version built with VC 2010 or GCC 4.5.2 works fine.  
The workaround is adding a volatile qualifier.

Additionally, I found some problems in Range#bsearch method.

1. The following example in documentation is wrong.  
(0..4).bsearch {|i| 100 - i } #=> 1, 2 or 3  
(0..4).bsearch {|i| 300 - i } #=> nil  
(0..4).bsearch {|i| 50 - i } #=> nil
2. (0.0...Float::INFINITY).bsearch {|x| Math.log(x) >= 0 } returns nil instead of 1.0 on Windows.
3. (0.0..10).bsearch {|x| 7.0-x} returns nil instead of 7.0

Here is a patch for all above problems:

```
diff --git a/range.c b/range.c.new
index 7d30383..0e0fd14 100644
--- a/range.c
+++ b/range.c.new
@@ -513,9 +513,9 @@ range_step(int argc, VALUE *argv, VALUE range)


- satisfies the condition, it returns nil. *
- ary = [0, 100, 100, 100, 200]
  - * (0..4).bsearch {|i| 100 - i } #=> 1, 2 or 3
  - * (0..4).bsearch {|i| 300 - i } #=> nil
  - * (0..4).bsearch {|i| 50 - i } #=> nil
  - * (0..4).bsearch {|i| 100 - ary[i] } #=> 1, 2 or 3
  - * (0..4).bsearch {|i| 300 - ary[i] } #=> nil
  - * (0..4).bsearch {|i| 50 - ary[i] } #=> nil *
- You must not mix the two modes at a time; the block must always
- return either true/false, or always return a number. It is @@ -543,10 @@ range_bsearch(VALUE range) smaller = 0; \ } \ else if (rb_obj_is_kind_of(v, rb_cNumeric)) { \
  - switch (rb_cmpint(rb_funcall(v, id_cmp, 1, INT2FIX(0)), v, INT2FIX(0)) < 0) { \
  - switch (rb_cmpint(rb_funcall(v, id_cmp, 1, INT2FIX(0)), v, INT2FIX(0))) { \ case 0: return val; \
  - case 1: smaller = 1; \
  - case -1: smaller = 0; \
  - case -1: smaller = 1; break; \
  - case 1: smaller = 0; \ } \ } \ else { @@ -586,6 +586,7 @@ range_bsearch(VALUE range) double high = RFLOAT_VALUE(rb_Float(end)); double mid, org_high; int count;
  - org_high = high; #ifdef FLT_RADIX #ifndef DBL_MANT_DIG #define BSEARCH_MAXCOUNT (((FLT_RADIX) - 1) * (DBL_MANT_DIG + DBL_MAX_EXP) + 100) @@ -646,7 +647,7 @@ range_bsearch(VALUE range) } if (isinf(low) && low < 0) { /* the range is (-INFINITY..high) */
  - double nlow = -1.0, dec;
  - volatile double nlow = -1.0, dec; if (nlow > high) nlow = high; count = BSEARCH_MAXCOUNT; /* find lower bound by checking low, low*2, low*4, ... / @@ -697,7 +698,6 @@ range_bsearch(VALUE range) binsearch: / find the desired value within low..high // where low is not -INFINITY and high is not INFINITY */
  - org_high = high; count = BSEARCH_MAXCOUNT; while (low < high && count >= 0) { mid = low + ((high - low) / 2);

```

### #3 - 11/15/2012 10:50 PM - mame (Yusuke Endoh)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset [r37662](#).

Luis, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

- range.c (range\_bsearch): fix some bugs: a documentation bug, a wrong condition, missed break in switch/case, and workaround for GCC optimization. See [ruby-core:49364] in detail. A great patch from Heesob Park. [Bug #7352] [Feature #4766]
- array.c (rb\_ary\_bsearch): fix similar bug (missed break).
- test/ruby/test\_range.rb: add two test cases for above.

**#4 - 11/15/2012 11:20 PM - mame (Yusuke Endoh)**

phasis68 (Heesob Park) wrote:

I guess this bug is related with the optimization of GCC 4.7.2.  
The version built with VC 2010 or GCC 4.5.2 works fine.  
The workaround is adding a volatile qualifier.

Additionally, I found some problems in Range#bsearch method.

I've committed your patch, my MEGA thanks!

I would really like you to have a commit bit. Are you willing?

Matz, what do you thing? He is a great all-round player; his contribution to Ruby includes deadlock issues (his patches was applied by kosaki), windows issues (accepted by usa and naruse), algorithmic issue (accepted by mrkn and me), documentation issues (by nobu), etc.

```
$ grep -i heesob ChangeLog doc/ChangeLog-* | wc -l
18
```

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#5 - 11/16/2012 12:14 AM - phasis68 (Heesob Park)**

Thanks, but I don't want to have a commit bit.

I am content with a tester and debugger.

**#6 - 11/16/2012 07:55 AM - mame (Yusuke Endoh)**

Okay. Tell me if you change your mind!

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)