Hi,

Trying to do `int*some_large_number` can result in
"warning: in a*b, b may be too big"
#=> Infinity

By example,
```
2^7830457
```

It is not good behavior, as we can make a workaround and it works perfectly:
```
# 7830457 = 2 * 37 * 105817 - 1
n = ((2 ** 105817) ** 37) ** 2 / 2 # => ...8739992577 which is the good number
```

I think it is not consistent to return Infinity when a correct answer can be given.
And Integer**Integer should always return an Integer (Infinity is a Float)

Also, while doing some tests about this I noticed that "bignum == Float::INFINITY" or "bignum.infinite?" hangs, while it should not, as it is always false.

See the script attached for some details and a (nonsense) implementation Integer#int_pow method which show it is possible to get better results.

Associated revisions
Revision ed4c5f38 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35081 b2dd03c8-39d4-4d8f-98ff-823fe69b080e
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)
```
• bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]
```
bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]

Revision 35081 - 03/18/2012 08:17 AM - nobu (Nobuyoshi Nakada)

bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]

Revision 49dbacf7 - 06/26/2012 07:21 PM - naruse (Yui NARUSE)
merge revision(s) 35081: [Backport #6605]
*b bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]*

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@36226 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 06/11/2010 10:37 PM - Eregon (Benoit Daloze)
*begin*
"And Integer* Integer should always return an Integer (Infinity is a Float)"
=> I meant Integer*(positive Integer) of course
*end*

#2 - 06/11/2010 10:58 PM - mame (Yusuke Endoh)
- Target version changed from 1.9.2 to 2.0.0

*begin*
Hi,
I agree that it is not good behavior, but it is definitely intended and traditional behavior. Not a bug.
1.9.2 should not be changed. Let's discuss towards 1.9.3. I move to 1.9.x feature request.
--
Yusuke Endoh mame@tsg.ne.jp
*end*

#3 - 06/11/2010 11:45 PM - Eregon (Benoit Daloze)
*begin*
On 11 June 2010 15:58, Yusuke Endoh redmine@ruby-lang.org wrote:
| I agree that it is not good behavior, but it is definitely intended and traditional behavior. Not a bug.
| For a mathematician, and even if he takes care of computing limitations, he would think it is a bug I believe ;)
| Python, by example, does not complain (and it works) for this. But yes, this behavior is common in many languages I suppose, but as we have Bignum built-in, it seems logical Integer*(Integer>0) returns an Integer, whatever the size is, at least until it becomes too bad for the memory.
| 1.9.2 should not be changed. Let's discuss towards 1.9.3. I move to 1.9.x feature request.
| Sure, I was hesitating for the target. As I think it is important because it is basic maths, I did choose 1.9.2.
| Yusuke Endoh mame@tsg.ne.jp
| B.D
*end*

#4 - 03/18/2012 04:50 PM - nahi (Hiroshi Nakamura)
- Description updated
- Assignee set to mrkn (Kenta Murata)
mrkn, please handle this.

#5 - 03/18/2012 05:17 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r35081.
Benoit, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- bignum.c (rb_big_pow): estimate result bit size more precisely. [ruby-core:30735][Feature #3429]

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer_pow.rb</td>
<td>1.23 KB</td>
<td>06/11/2010</td>
<td>Eregon (Benoit Daloze)</td>
</tr>
</tbody>
</table>