Ruby cannot handle NaN as a unique key of Hash.

Here is an example:

```ruby
irb(main):001:0> h = {}
=> {}  
irb(main):002:0> h[0/0.0]=1
=> 1  
irb(main):003:0> h[0/0.0]=2
=> 2  
irb(main):004:0> h[0/0.0]=3
=> 3  
irb(main):005:0> h
=> {NaN=>1, NaN=>2, NaN=>3}  
irb(main):006:0> h[0/0.0]
=> nil
```

I think this is related with the NaN comparing problem:

```ruby
irb(main):001:0> 0/0.0 == 0/0.0
=> false
irb(main):002:0> a = 0/0.0
=> NaN
irb(main):003:0> a == a
=> false
irb(main):004:0> a <=> a
=> nil
```

Associated revisions

Revision 52372 - 10/30/2015 12:54 AM - hsbt (Hiroshi SHIBATA)

- lib/rubygems: Update to RubyGems HEAD(60d7972). this version contains pull requests number of #1343, #1356, #1357, #1363 at https://github.com/rubygems/rubygems/pulls
- test/rubygems: ditto.

Revision 52372 - 10/30/2015 12:54 AM - hsbt (Hiroshi SHIBATA)

- lib/rubygems: Update to RubyGems HEAD(60d7972). this version contains pull requests number of #1343, #1356, #1357, #1363 at https://github.com/rubygems/rubygems/pulls
- test/rubygems: ditto.

Revision 52372 - 10/30/2015 12:54 AM - hsbt (Hiroshi SHIBATA)

- lib/rubygems: Update to RubyGems HEAD(60d7972). this version contains pull requests number of #1343, #1356, #1357, #1363 at https://github.com/rubygems/rubygems/pulls
- test/rubygems: ditto.
Revision 52372 - 10/30/2015 12:54 AM - hsbt (Hiroshi SHIBATA)

- lib/rubygems: Update to RubyGems HEAD(60d7972). this version contains pull requests number of #1343, #1356, #1357, #1363 at https://github.com/rubygems/rubygems/pulls
- test/rubygems: ditto.

Revision 52372 - 10/30/2015 12:54 AM - hsbt (Hiroshi SHIBATA)

- lib/rubygems: Update to RubyGems HEAD(60d7972). this version contains pull requests number of #1343, #1356, #1357, #1363 at https://github.com/rubygems/rubygems/pulls
- test/rubygems: ditto.

History

#1 - 05/11/2009 07:04 PM - nobu (Nobuyoshi Nakada)

=begin
Hi,

At Wed, 8 Apr 2009 14:38:31 +0900, Heesob Park wrote in [ruby-core:23154]:

Ruby cannot handle NaN as a unique key of Hash.

It's easy to make them unique as key, I'm not sure which is "correct" behavior though.

I think this is related with the NaN comparing problem:

```ruby
irb(main):001:0> 0/0.0 == 0/0.0
=> false
irb(main):002:0> a = 0/0.0
=> NaN
irb(main):003:0> a == a
=> false
irb(main):004:0> a <=> a
=> nil
```

I think it's mathematically correct behavior, and different thing from Hash.

Index: numeric.c

--- numeric.c (revision 23390)
+++ numeric.c (working copy)
@@ -1127,7 +1127,6 @@ flo_eql(VALUE x, VALUE y)
 double a = RFLOAT_VALUE(x);
 double b = RFLOAT_VALUE(y);
-#if defined(_MSC_VER) && _MSC_VER < 1300
if (isnan(a) || isnan(b)) return Qfalse; -#endif
if (isnan(a) && isnan(b))
  return memcmp(&a, &b, sizeof(a)) ? Qfalse : Qtrue; if (a == b)     return Qtrue;
--
Nobu Nakada
=end

#2 - 05/11/2009 09:21 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:23423] Re: [Bug #1363] Wrong value for Hash of NaN" on Mon, 11 May 2009 19:03:59 +0900, Nobuyoshi Nakada nobu@ruby-lang.org writes:

[At Wed, 8 Apr 2009 14:38:31 +0900, Heesob Park wrote in [ruby-core:23154]:

> Ruby cannot handle NaN as a unique key of Hash.
> It's easy to make them unique as key, I'm not sure which is
"correct" behavior though.

Two NaN values are not equal with each other by definition, in that sense, even though some might want to use NaN as a hash key, the current behavior seems to be "correct".

Yukihiro Matsumoto wrote:

Hi,

In message "Re: [ruby-core:23423] Re: [Bug #1363] Wrong value for Hash of NaN" on Mon, 11 May 2009 19:03:59 +0900, Nobuyoshi Nakada nobu@ruby-lang.org writes:

|At Wed, 8 Apr 2009 14:38:31 +0900, [Heesob Park wrote in [ruby-core:23154]:
|> Ruby cannot handle NaN as a unique key of Hash.
| |
|It's easy to make them unique as key, I'm not sure which is "correct" behavior though.

Two NaN values are not equal with each other by definition, in that sense, even though some might want to use NaN as a hash key, the current behavior seems to be "correct".

I don't argue about correctness on NaNs being different each other, but I also agree with the reporter with Hash being inconvenient when NaNs used as keys.

Attachment: signature.asc

Yukihiro Matsumoto wrote:

Hi,

In message "Re: [ruby-core:23427] Re: [Bug #1363] Wrong value for Hash of NaN" on Mon, 11 May 2009 21:58:42 +0900, Urabe Shyouhei shyouhei@ruby-lang.org writes:

|I don't argue about correctness on NaNs being different each other, but I also agree with the reporter with Hash being inconvenient when NaNs used as keys.

Using float values as hash keys should be strongly discouraged, because you may not have expected result anyway. A 3.14 may be different in bit-wise representation from another 3.14. I have no plan to prohibit using them as keys, right now, but see no need to change the language to encourage bad habit. If I have to make a change related to this issue, I'd rather prohibit NaN as a key (or float keys in general), as in [ruby-core:23426].
The behavior OP reported is not a bug. So I close the ticket.

Indeed, the behavior is confusing a little. IMO, it would be good to warn when Float is used as hash keys. Anyway, warning or prohibiting NaN is a new feature, not bug fix for this ticket. If you still want, please register a new ticket to Feature tracker.

Thanks,

--
Yusuke Endoh mame@tsg.ne.jp