

Ruby - Bug #11379

Difference in `Array#sort\_by` method between Ubuntu Linux and OS X

07/21/2015 09:37 AM - pibako (Piotr Kowalski)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.3.0dev (2015-07-21 trunk 51310) [i686-linux]	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN
Description The sort order (for the same objects) is different depending on the operating system. It looks like the Linux version of sorting has changed versus the previous ruby versions. On Linux: <pre>piotr@pi-virtualbox:~\$ uname -a Linux pi-virtualbox 3.13.0-49-generic #83-Ubuntu SMP Fri Apr 10 20:14:51 UTC 2015 i686 i686 i686 GNU/Linux piotr@pi-virtualbox:~\$ ruby --version ruby 2.2.2p95 (2015-04-13 revision 50295) [i686-linux] piotr@pi-virtualbox:~\$ irb irb(main):001:0> ['az', 'ac', 'AC'].sort_by { s s.downcase} => ["ac", "AC", "az"]</pre> On OS X: <pre>pi@mac:~\$ uname -a Darwin mac.local 14.4.0 Darwin Kernel Version 14.4.0: Thu May 28 11:35:04 PDT 2015; root:xnu-2782.30.5~1/RELEASE_X86_64 x86_64 pi@mac:~\$ ruby --version ruby 2.2.2p95 (2015-04-13 revision 50295) [x86_64-darwin14] pi@mac:~\$ irb irb(main):001:0> ['az', 'ac', 'AC'].sort_by { s s.downcase} => ["AC", "ac", "az"]</pre> The bug is not present in 1.9.3 and 2.1.x. I also confirmed it's not fixed in 2.2.2 and 2.3.0-dev.		

History

#1 - 07/21/2015 10:06 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

Not a bug.

Array#sort_by is not stable sort.

#2 - 07/21/2015 10:13 AM - nobu (Nobuyoshi Nakada)

- Status changed from Rejected to Open

Sorry, it is not a stability issue.

#3 - 07/21/2015 10:28 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

It is a stability issue.

Enumerable#sort_by compares only the results of the given block, so can't differentiate "ac" and "AC".

Maybe you may want to write as:

```
['az', 'ac', 'AC'].sort_by {|s| [s.downcase, s]}
```

#4 - 07/21/2015 10:36 AM - pibako (Piotr Kowalski)

What do you mean that it's a stability issue? I understand that the order cannot be guaranteed but I would expect it to work the same on all platforms.

Also this code works fine with 1.9.3 and 2.1.6 and only fails with 2.2.

#5 - 07/22/2015 12:50 PM - funny_falcon (Yura Sokolov)

I understand that the order cannot be guaranteed but I would expect it to work the same on all platforms.

And you will be wrong.

You may not expect order is same even with two subsequent sort invocation.

Every "same order" produced by unstable sorting on a set with some equal elements should be treated as occasional.

#6 - 07/22/2015 01:36 PM - pibako (Piotr Kowalski)

You may not expect order is same even with two subsequent sort invocation.

Non deterministic algorithm... makes sense. Is that somewhere stated in documentation?

Just curious, do you happen to know what has changed in ruby 2.2 linux version?

#7 - 07/22/2015 01:39 PM - nobu (Nobuyoshi Nakada)

Because `qsort_r()` of glibc is used instead of `ruby_qsort()`.

#8 - 07/22/2015 01:43 PM - pibako (Piotr Kowalski)

All clear, thank you.