

Ruby - Bug #15174

Sorting array of elements using "&:size" block, in which more than 1 element

09/28/2018 09:30 PM - Mr_Cartoon (Mr Cartoon)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	2.5.1, 2.6.0-preview2	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description Example with ranges(first with what defined bug), but also tested on strings, symbols, integers Good <pre>[1..2].sort { x x.size } [1..2, 4..5].sort { x x.size } [1..2].sort(&:size)</pre> Bad <pre>[1..2, 4..5].sort(&:size)</pre> Traceback (most recent call last): 4: from /home/sviat/.rvm/rubies/ruby-2.6.0-preview2/bin/irb:11:in <main>' 3: from (irb):3 2: from (irb):3:in sort' 1: from (irb):3:in `size' ArgumentError (wrong number of arguments (given 1, expected 0)) Tested ruby versions: 2.5.1, 2.6.0-preview2		

History

#1 - 09/28/2018 10:25 PM - jeremyevans0 (Jeremy Evans)

You want sort_by, not sort:

```
[1..2, 4..5].sort_by(&:size)  
# => [1..2, 4..5]
```

The difference in behavior between { |x| x.size } and (&:size) is due to how ruby handles Symbol#to_proc when multiple arguments are yielded. If a block yields multiple arguments, Symbol#to_proc is the equivalent of first_arg.send(symbol, *remaining_args), not first_arg.send(symbol) or [first_arg, *remaining_args].send(symbol). Example:

```
def b(*a)  
  p [self, a]  
  1  
end  
  
[1..2, 4..5].sort{|x| x.b}  
# => [4..5, 1..2]  
# output:  
# [1..2, []]  
  
[1..2, 4..5].sort(&b)  
# => [4..5, 1..2]  
# output:  
# [1..2, [4..5]]
```

Range#size does not take an argument, hence the ArgumentError when you are using sort(&:size) with an array of Range instances.

I believe this behavior is expected (see vm_yield_with_symbol in vm_insnhelper.c). However, a more experienced rubyist should probably confirm that.

#2 - 09/28/2018 10:45 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

An expected behavior.