

Ruby - Feature #16995

Sets: <=> should be specialized

06/26/2020 08:43 PM - marcandre (Marc-Andre Lafortune)

<div>Status:Closed</div> <div>Priority:Normal</div> <div>Assignee:</div> <div>Target version:</div>	
<div>Description</div> <div>This is quite minor, but Set#<=> should be refined.</div> <div>Reminder: Set defines <, >, etc. as inclusion, but does not have a corresponding <=>:</div> <div>Set[1] < Set[1, 2] # => true</div> <div>Set[1] <=> Set[1, 2] # => nil, should be -1</div> <div>Set[1] <=> Set[2] # => nil, ok, not orderable</div> <div>The official stated reason for Set to <i>not</i> implement is that some sets are not comparable. That is exactly what nil result type is for IMO. Sets are partially ordered and <=> should reflect that. https://en.wikipedia.org/wiki/Partially_ordered_set</div> <div>Set[1] < Set[1, 2] # => true</div> <div>[Set[1], Set[1, 2]].sort # => ArgumentError, should be [Set[1], Set[1, 2]]</div> <div>[Set[1], Set[2]].sort # => ArgumentError, ok, can't be ordered</div> <div>This is <i>exactly the same</i> idea as Class, which correctly refines <=>:</div> <div>Array < Enumerable # => true</div> <div>Array <=> Enumerable # => -1, ok</div> <div>[Array, Enumerable].sort # => [Array, Enumerable]</div> <div>[Array, String].sort # => ArgumentError (comparison of Class with Class failed), ok</div>	
<div>Related issues:</div> <div>Related to Ruby - Feature #16989: Sets: need ♥️</div> <div>Assigned</div>	

History

#1 - 06/26/2020 08:47 PM - marcandre (Marc-Andre Lafortune)

- Related to Feature #16989: Sets: need ♥️ added

#2 - 01/01/2023 01:04 PM - zverok (Victor Shepelev)

- Status changed from Open to Closed

Implemented in Ruby 3.0:

```
Set[1] <=> Set[1, 2]
# => -1
[Set[1], Set[1, 2]].sort
# => [#<Set: {1}>, #<Set: {1, 2}>]
```