Ruby - Feature #6589

Set#rehash

06/14/2012 11:54 AM - marcandre (Marc-Andre Lafortune)

Status:	Closed	
Priority:	Normal	
Assignee:	knu (Akinori MUSHA)	
Target version:	2.6	

Description

There should be a way to rehash a Set.

```
s = Set.new([[]])
s.first << 1
# s.rehash # Does not exist!
s.include? [1] # => false, want true
```

See also:

http://stackoverflow.com/questions/10992423/is-this-expected-behaviour-for-a-set-of-arrays-in-ruby
http://stackoverflow.com/questions/10361400/deleting-a-modified-object-from-a-set-in-a-no-op

Related issues:

Related to Ruby - Bug #12970: == Equality of recursive sets fails	**Closed**

Associated revisions

Revision 8c90432af72a59d8934b2c94a1bc847449e1f393 - 10/22/2017 12:25 PM - Akinori MUSHA

Add Set#reset

This method resets the internal state of a set after modification to
existing elements, reindexing and deduplicating them. [Feature #6589]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60360 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8c90432a - 10/22/2017 12:25 PM - Akinori MUSHA

Add Set#reset

This method resets the internal state of a set after modification to
existing elements, reindexing and deduplicating them. [Feature #6589]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60360 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 07/14/2012 06:35 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

#2 - 10/25/2012 07:44 PM - yhara (Yutaka HARA)

- Target version changed from 2.0.0 to 2.6

#3 - 11/12/2012 01:12 PM - marcandre (Marc-Andre Lafortune)

Comment about this trivial but needed feature would be appreciated.

#4 - 11/25/2012 02:05 AM - headius (Charles Nutter)

Is it specified that Set must be hashtable-based forever? There are alternate ways to implement a Set.

#5 - 11/25/2012 11:17 AM - marcandre (Marc-Andre Lafortune)

headius (Charles Nutter) wrote:

> Is it specified that Set must be hashtable-based forever? There are alternate ways to implement a Set.

Alternate ways of implementing Set with check/insertion in O(1) that would also work if structures change without a rehash functionality?

In any case, the documentation states that "Set uses Hash as storage", but more importantly that "The equality of each couple of elements is determined according to Object#eql? and Object#hash".

#6 - 07/27/2013 03:48 PM - knu (Akinori MUSHA)

Actually, an undocumented "feature" is that Set does not support an element being modified once it is added.

Maybe we should "clarify" that in the document, or add such a method that recalculates identities of elements. I'm yet to decide which, and the name we could give it.

- rehash (let's be honest)
- reset (re-set the set)
- sync
- ...

#7 - 07/31/2013 03:13 PM - knu (Akinori MUSHA)

- Status changed from Assigned to Feedback

I added some notes to the rdoc in r42265.

#8 - 11/25/2016 10:05 PM - marcandre (Marc-Andre Lafortune)

- Related to Bug #12970: == Equality of recursive sets fails added

#9 - 10/22/2017 12:25 PM - knu (Akinori MUSHA)

- Status changed from Feedback to Closed

Applied in changeset trunk|r60360.

Add Set#reset

This method resets the internal state of a set after modification to
existing elements, reindexing and deduplicating them. [Feature #6589]