**Ruby master - Feature #6588**

**Set#intersect?**

06/14/2012 11:47 AM - marcandre (Marc-Andre Lafortune)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>knu (Akinori MUSHA)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.6</td>
</tr>
</tbody>
</table>

**Description**

There is Set#superset?, Set#subset? with their proper variants, but there is no Set#intersect? nor Set#disjoint?

To check if two sets intersect, one can do

```ruby
set.intersection(other).empty?
```

This cycles through all elements, though. To be efficient, one can instead do the iteration manually:

```ruby
other.any? { |x| set.include?(x) }
```

I think it would be natural to add Set#intersect? and its reverse Set#disjoint?

```ruby
class Set
  def intersect?(enum)
    enum.any? { |x| include?(x) }
  end
end
```

Maybe it would be worth it to optimize it if enum is a larger Set by starting it with

```ruby
  return any? { |x| enum.include?(x) } if enum.size > size
```

**Associated revisions**

- Revision bd304ed8 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
  Add Set#intersect? and #disjoint?.

  - `lib/set.rb` (Set#intersect?, Set#disjoint?): Add new methods for testing if two sets have any element in common. [ruby-core:45641] [Feature #6588] Based on the code by marcandre.

  git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@42253 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

- Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
  Add Set#intersect? and #disjoint?.

  - `lib/set.rb` (Set#intersect?, Set#disjoint?): Add new methods for testing if two sets have any element in common. [ruby-core:45641] [Feature #6588] Based on the code by marcandre.

- Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
  Add Set#intersect? and #disjoint?.

  - `lib/set.rb` (Set#intersect?, Set#disjoint?): Add new methods for testing if two sets have any element in common. [ruby-core:45641] [Feature #6588] Based on the code by marcandre.

- Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
  Add Set#intersect? and #disjoint?.

  - `lib/set.rb` (Set#intersect?, Set#disjoint?): Add new methods for testing if two sets have any element in common. [ruby-core:45641] [Feature #6588] Based on the code by marcandre.

- Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
  Add Set#intersect? and #disjoint?.

  - `lib/set.rb` (Set#intersect?, Set#disjoint?): Add new methods for testing if two sets have any element in common. [ruby-core:45641] [Feature #6588] Based on the code by marcandre.

- Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
  Add Set#intersect? and #disjoint?.
Add new methods for testing if two sets have any element in common.

Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
Add Set#intersect? and #disjoint?.

Revision 42253 - 07/30/2013 09:58 AM - knu (Akinori MUSHA)
Add Set#intersect? and #disjoint?.

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 these simple features would be appreciated.

#4 - 11/12/2012 06:08 PM - alexeymuranov (Alexey Muranov)
+1. Maybe #meet? instead of #intersect? ? It can be argued that any set intersects any other, just the intersection is sometimes empty :).

#5 - 11/13/2012 12:38 AM - marcandre (Marc-Andre Lafortune)
alexeymuranov (Alexey Muranov) wrote:

+1.

Thanks for the +1

It can be argued that any set intersects any other, just the intersection is sometimes empty :).

No, I believe it would be wrong to argue that. From wikipedia: "We say that A intersects B if A intersects B at some element"

Moreover: "We say that A and B are disjoint if A does not intersect B. In plain language, they have no elements in common"

I believe that both intersect? and disjoint? are the established terms for the concept I'm proposing.

#6 - 07/27/2013 02:59 PM - knu (Akinori MUSHA)
OK, accepted. I'll work on it.

#7 - 07/30/2013 06:58 PM - knu (Akinori MUSHA)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r42253.
Marc-Andre, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.
I followed superset?() and the like, and made the new methods accept only a set for the moment, because I couldn't come up with an idea of how to deal with Range. For example:

- if Set[2].intersect?(1.5..2.5) should return true
- if Set[2].intersect?(3..(1.0/0)) should immediately return false using some knowledge on Range