I propose to define Array#max. It is 10+ times faster than Enumerable#max since it skips a call to #each.

```ruby
a = [*1..10000]; 100000.times { a.max }
```

- no patch: 22.424s
- Array#max defined: 1.740s

I don't think it is a good idea to copy all Enumerable methods to Array. But there are two reasons why max is special:

- It is one of the most basic operations for big data processing.
- We often use an idiom `[a, b].max` because of the lack of Math.max(a, b).

I think the latter is particularly important. The idiom is concise but unsuitable in a hotspot since it creates a temporal array. If Array#max is defined, we can easily optimize the idiom by introducing a special instruction like opt_newarray_max.

```ruby
x, y = 1, 2; 10000000.times { [x, y].max }
```

- no patch: 2.799s
- Array#max defined: 1.224s
- opt_newarray_max: 0.555s

The patches are attached. (0001 is a preparation. 0002 introduces Array#max. 0003 introduces a special instruction.)

Of course, we can say the same for Array#min. The patches include Array#min too.

What do you think?

Associated revisions
Revision 68a6f2e9 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)

- array.c (rb_ary_max, rb_ary_min): Array#max and Array#min added. [Feature #12172]
- internal.h (OPTIMIZED_CMP): moved from enum.c so that array.c can use it.
- test/ruby/test_array.rb (test_max, test_min): tests for Array#max
and Array#min.

- test/ruby/test_enum.rb (test_max, test_min): revised a bit to test
  Enumerable#max and #min explicitly.

Revision 54150 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)

- array.c (rb_ary_max, rb_ary_min): Array#max and Array#min added.
  [Feature #12172]

- internal.h (OPTIMIZED_CMP): moved from enum.c so that array.c can
  use it.

- test/ruby/test_array.rb (test_max, test_min): tests for Array#max
  and Array#min.

- test/ruby/test_enum.rb (test_max, test_min): revised a bit to test
  Enumerable#max and #min explicitly.

Revision 54150 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)

Revision 54150 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)

Revision 54150 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)

Revision 54150 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)
- test/ruby/test_array.rb (test_max, test_min): tests for Array#max and Array#min.
- test/ruby/test_enum.rb (test_max, test_min): revised a bit to test Enumerable#max and #min explicitly.

Revision f8e29640 - 03/17/2016 12:49 PM - mame (Yusuke Endoh)

- NEWS: add Array#max, #min, and the optimization. [Feature #12172]

Revision 54154 - 03/17/2016 12:49 PM - mame (Yusuke Endoh)

- NEWS: add Array#max, #min, and the optimization. [Feature #12172]

Revision 54154 - 03/17/2016 12:49 PM - mame (Yusuke Endoh)

- NEWS: add Array#max, #min, and the optimization. [Feature #12172]

Revision 54154 - 03/17/2016 12:49 PM - mame (Yusuke Endoh)

- NEWS: add Array#max, #min, and the optimization. [Feature #12172]

Revision 54154 - 03/17/2016 12:49 PM - mame (Yusuke Endoh)

- NEWS: add Array#max, #min, and the optimization. [Feature #12172]

History

#1 - 03/14/2016 02:44 PM - mame (Yusuke Endoh)
- Description updated

#2 - 03/17/2016 11:58 AM - mame (Yusuke Endoh)
- Status changed from Open to Assigned
- Assignee set to mame (Yusuke Endoh)

Matz and ko1 accepted this proposal. I'll commit.

#3 - 03/17/2016 12:14 PM - mame (Yusuke Endoh)
- Status changed from Assigned to Closed

Applied in changeset r54150.

- array.c (rb_ary_max, rb_ary_min): Array#max and Array#min added. [Feature #12172]
- internal.h (OPTIMIZED_CMP): moved from enum.c so that array.c can use it.
- test/ruby/test_array.rb (test_max, test_min): tests for Array#max and Array#min.
- test/ruby/test_enum.rb (test_max, test_min): revised a bit to test Enumerable#max and #min explicitly.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001-refactor-a-data-structure-for-CMP_OPTIMIZABLE.patch</td>
<td>5.3 KB</td>
<td>03/14/2016</td>
<td>mame (Yusuke Endoh)</td>
</tr>
<tr>
<td>0002-introduce-Array-max-and-Array-min.patch</td>
<td>5.11 KB</td>
<td>03/14/2016</td>
<td>mame (Yusuke Endoh)</td>
</tr>
</tbody>
</table>