

Ruby trunk - Feature #12172

Array#max and Array#min

03/14/2016 01:30 PM - mame (Yusuke Endoh)

Status:	Closed
Priority:	Normal
Assignee:	mame (Yusuke Endoh)
Target version:	
Description	
<p>I propose to define Array#max. It is 10+ times faster than Enumerable#max since it skips a call to #each.</p>	
<pre>a = [*1..10000]; 100000.times { a.max }</pre>	
<ul style="list-style-type: none">• no patch: 22.424s• Array#max defined: 1.740s	
<p>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:</p>	
<ul style="list-style-type: none">• 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).	
<p>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.</p>	
<pre>x, y = 1, 2; 1000000.times { [x, y].max }</pre>	
<ul style="list-style-type: none">• no patch: 2.799s• Array#max defined: 1.224s• opt_newarray_max: 0.555s	
<pre>\$./miniruby --dump=insns -e 'x, y = 1, 2; [x, y].max'</pre>	
<pre>== disasm: #<ISeq:<main>@-e>=====</pre>	
<pre>local table (size: 3, argc: 0 [opts: 0, rest: -1, post: 0, block: -1, kw: -1@-1, kwrest: -1])</pre>	
<pre>[3] x [2] y</pre>	
<pre>0000 trace 1 (1)</pre>	
<pre>0002 putobject_OP_INT2FIX_O_1_C_</pre>	
<pre>0003 putobject 2</pre>	
<pre>0005 setlocal_OP_WC__0 2</pre>	
<pre>0007 setlocal_OP_WC__0 3</pre>	
<pre>0009 getlocal_OP_WC__0 3</pre>	
<pre>0011 getlocal_OP_WC__0 2</pre>	
<pre>0013 opt_newarray_max 2</pre>	
<pre>0015 leave</pre>	
<p>The patches are attached. (0001 is a preparation. 0002 introduces Array#max. 0003 introduces a special instruction.)</p>	
<p>Of course, we can say the same for Array#min. The patches include Array#min too.</p>	
<p>What do you think?</p>	

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.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@54150 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

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)

- 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)

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

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

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@54154 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

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

0001-refactor-a-data-structure-for-CMP_OPTIMIZABLE.patch	5.3 KB	03/14/2016	mame (Yusuke Endoh)
0002-introduce-Array-max-and-Array-min.patch	5.11 KB	03/14/2016	mame (Yusuke Endoh)

