

## Ruby master - Bug #13443

### Improve performance of Range#{min,max}

04/17/2017 05:30 AM - watson1978 (Shizuo Fujita)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b>	<b>Backport:</b> 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

#### Description

Range#{min,max} will be faster around 30%.

#### Before

	user	system	total	real
Range#min	1.270000	0.010000	1.280000 (	1.279449)
Range#max	1.300000	0.000000	1.300000 (	1.310150)

#### After

	user	system	total	real
Range#min	0.940000	0.010000	0.950000 (	0.967873)
Range#max	0.960000	0.010000	0.970000 (	0.983417)

#### Test code

```
require 'benchmark'

Benchmark.bmbm do |x|

  x.report "Range#min" do
    1000000.times do
      (1..100).min
    end
  end

  x.report "Range#max" do
    1000000.times do
      (1..100).max
    end
  end

end
```

#### Patch

<https://github.com/ruby/ruby/pull/1585>

#### Associated revisions

Revision ab3a40c1 - 05/30/2017 02:57 AM - watson1978 (Shizuo Fujita)

Improve performance of Range#{min,max}

range.c (range\_min): use OPTIMIZED\_CMP() to compare the objects instead of '<=>' method dispatching for Fixnum/Float/String object inside Range object.

range.c (range\_max): ditto.

Range#min -> 34 % up  
Range#max -> 44 % up

### Before

Range#min	8.428M	(± 1.3%)	i/s -	42.141M in	5.000952s
Range#max	8.157M	(± 1.3%)	i/s -	40.852M in	5.009297s

### After

Range#min	11.269M	(± 1.2%)	i/s -	56.388M in	5.004611s
Range#max	11.764M	(± 1.3%)	i/s -	58.856M in	5.003820s

### Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|
  x.report "Range#min" do |i|
    i.times { (1..100).min }
  end

  x.report "Range#max" do |i|
    i.times { (1..100).max }
  end
end
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58964 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 58964 - 05/30/2017 02:57 AM - watson1978 (Shizuo Fujita)

Improve performance of Range#{min,max}

range.c (range\_min): use OPTIMIZED\_CMP() to compare the objects instead of '<=>' method dispatching for Fixnum/Float/String object inside Range object.

range.c (range\_max): ditto.

Range#min -> 34 % up  
Range#max -> 44 % up

### Before

Range#min	8.428M	(± 1.3%)	i/s -	42.141M in	5.000952s
Range#max	8.157M	(± 1.3%)	i/s -	40.852M in	5.009297s

### After

Range#min	11.269M	(± 1.2%)	i/s -	56.388M in	5.004611s
Range#max	11.764M	(± 1.3%)	i/s -	58.856M in	5.003820s

### Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|
  x.report "Range#min" do |i|
    i.times { (1..100).min }
  end

  x.report "Range#max" do |i|
    i.times { (1..100).max }
  end
end
```

### Revision 58964 - 05/30/2017 02:57 AM - watson1978 (Shizuo Fujita)

Improve performance of Range#{min,max}

range.c (range\_min): use OPTIMIZED\_CMP() to compare the objects instead of '<=>' method dispatching for Fixnum/Float/String object inside Range object.

range.c (range\_max): ditto.

```
Range#min -> 34 % up
Range#max -> 44 % up
```

[ruby-core:80713] [Bug #13443] [Fix GH-1585]

## Before

Range#min	8.428M (± 1.3%)	i/s -	42.141M in	5.000952s
Range#max	8.157M (± 1.3%)	i/s -	40.852M in	5.009297s

## After

Range#min	11.269M (± 1.2%)	i/s -	56.388M in	5.004611s
Range#max	11.764M (± 1.3%)	i/s -	58.856M in	5.003820s

## Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|
  x.report "Range#min" do |i|
    i.times { (1..100).min }
  end

  x.report "Range#max" do |i|
    i.times { (1..100).max }
  end
end
```

### Revision 58964 - 05/30/2017 02:57 AM - watson1978 (Shizuo Fujita)

Improve performance of Range#{min,max}

range.c (range\_min): use OPTIMIZED\_CMP() to compare the objects instead of '<=>' method dispatching for Fixnum/Float/String object inside Range object.

range.c (range\_max): ditto.

```
Range#min -> 34 % up
Range#max -> 44 % up
```

[ruby-core:80713] [Bug #13443] [Fix GH-1585]

## Before

Range#min	8.428M (± 1.3%)	i/s -	42.141M in	5.000952s
Range#max	8.157M (± 1.3%)	i/s -	40.852M in	5.009297s

## After

Range#min	11.269M (± 1.2%)	i/s -	56.388M in	5.004611s
Range#max	11.764M (± 1.3%)	i/s -	58.856M in	5.003820s

## Test code

```
require 'benchmark/ips'

Benchmark.ips do |x|
  x.report "Range#min" do |i|
    i.times { (1..100).min }
  end

  x.report "Range#max" do |i|
    i.times { (1..100).max }
  end
end
```

## History

### #1 - 05/30/2017 02:57 AM - watson1978 (Shizuo Fujita)

- Status changed from Open to Closed

Applied in changeset [trunk|r58964](#).

---

## Improve performance of Range#{min,max}

range.c (range\_min): use OPTIMIZED\_CMP() to compare the objects instead of '<=>' method dispatching for Fixnum/Float/String object inside Range object.

range.c (range\_max): ditto.

Range#min -> 34 % up  
Range#max -> 44 % up

[ruby-core:80713] [Bug #13443] [Fix GH-1585]

### Before

Range#min	8.428M	(± 1.3%)	i/s	-	42.141M	in	5.000952s
Range#max	8.157M	(± 1.3%)	i/s	-	40.852M	in	5.009297s

### After

Range#min	11.269M	(± 1.2%)	i/s	-	56.388M	in	5.004611s
Range#max	11.764M	(± 1.3%)	i/s	-	58.856M	in	5.003820s

### Test code

```
require 'benchmark/ips'
```

```
Benchmark.ips do |x|  
  x.report "Range#min" do |i|  
    i.times { (1..100).min }  
  end
```

```
  x.report "Range#max" do |i|  
    i.times { (1..100).max }  
  end  
end
```