

Ruby trunk - Bug #15370

Array#=== in long running loop continually consumes more RAM

12/03/2018 03:11 AM - danielpclark (Daniel P. Clark)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.5.3p105 (2018-10-18 revision 65156) [x86_64-linux]	Backport: 2.4: UNKNOWN, 2.5: UNKNOWN

Description

I was doing a code challenge and trying to use as few characters as possible (not shown). So instead of using `Array#member?` I tried `Array#===` and it consumed my system's RAM once I switched to that.

The good code here:

```
freq = [0]

$<.readlines.cycle do |num|
  sum = freq.last + num.to_i

  break sum if freq.member? sum

  freq << sum
end
```

Screenshot from 2018-12-02 21-33-38.png

Doesn't grow or consume any extra memory... which is good.

But when I change it to:

```
freq = [0]

$<.readlines.cycle do |num|
  sum = freq.last + num.to_i

  break sum if freq === sum

  freq << sum
end
```

It doesn't take long to fill up 10.7 Gigs and continue consuming more RAM.

Screenshot from 2018-12-02 21-41-38.png

To duplicate this behavior provide the following as input:

```
"-10\n+18\n+5\n+1\n+1\n-19\n-13\n-4\n-4\n-5\n-17\n+13\n-3\n+18\n-17\n+14\n+11\n+14\n+16\n+6\n-2\n-3\n-3\n-5\n+18\n+17\n-9\n+14\n+13\n+3\n-17\n-3\n+11\n-4\n-5\n+3\n+9\n-20\n-18\n+10\n+1\n+1\n-14\n+9\n-5\n+20\n-6\n+1\n+2\n+2\n-13\n-11\n+18\n+18\n+1\n+2\n-11\n+23\n+2\n-7\n+2\n+7\n-3\n+6\n+9\n+8\n+13\n-8\n+12\n+7\n+4\n+9\n-11\n-19\n+1\n+13\n+15\n-14\n+10\n+19\n-6\n+1\n+7\n-14\n+3\n+17\n+2\n+2\n+15\n+15\n+18\n+6\n-9\n+8\n+15\n+12\n+15\n+5\n+8\n+1\n+6\n-13\n+18\n+10\n+10\n+7\n+14\n+4\n+9\n+17\n+10\n+6\n+7\n+9\n+5\n+1\n+1\n+19\n+10\n+18\n-5\n+10\n+7\n-6\n-10\n+17\n-12\n-15\n-17\n+5\n+7\n-11\n+18\n-12\n-12\n-3\n+10\n-4\n-5\n+13\n+10\n-1\n+7\n-12\n-11\n-9\n-18\n+3\n-1\n-15\n-7\n-15\n+4\n-7\n-4\n-7\n+19\n-3\n+14\n+9\n-11\n-7\n+16\n-10\n-3\n+10\n+13\n+10\n-7\n+13\n+11\n+17\n+7\n+4\n+20\n+19\n-1\n-5\n+10\n+7\n-9\n-14\n-3\n-15\n+5\n-19\n+13\n+9\n-12\n-6\n-13\n-15\n-2\n-15\n-3\n+11\n+22\n+19\n+18\n-6\n+10\n-9\n+10\n+19\n-15\n-2\n+20\n+12\n-20\n-8\n+23\n+15\n-11\n+10\n-2\n+8\n-4\n-19\n+16\n+16\n-6\n-22\n-2\n+9\n+13\n-19\n+8\n+16\n+18\n+21\n+17\n+26\n-19\n+16\n+8\n-14\n-1\n+8\n+9\n+2\n+6\n+4\n-7\n-2\n+1\n+14\n-3\n+17\n+6\n+19\n-14\n+10\n+13\n-17\n-10\n+11\n-14\n-13\n-13\n+11\n+19\n+18\n+1\n-12\n+14\n-17\n+8\n+11\n+1\n+15\n+13\n-12\n+14\n+2\n-8\n+9\n+9\n+2\n+14\n+7\n+14\n-1\n-11\n+13\n-11\n-3\n-18\n+7\n-19\n-1\n-14\n+4\n-11\n+12\n-6\n-19\n+6\n+15\n-10\n-9\n+24\n-9\n-1\n+14\n+8\n-5\n-5\n+12\n+16\n-14\n+3\n-6\n+12\n+11\n-18\n+9\n+17\n+10\n+11\n-19\n+13\n+9\n+4\n+15\n+8\n-21\n+5\n-12\n+4\n+13\n+13\n+6\n+1\n+6\n+4\n+6\n-14\n-6\n+17\n-4\n+10\n+13\n+17\n+3\n-5\n+19
```

```
\n+6\n-16\n-8\n+21\n-19\n+4\n-15\n-5\n+7\n-11\n-19\n+18\n-2\n+10\n-17\n-8\n-8\n-10\n+7\n-19\n-6\n-11\n+19\n+5\n+9\n-11\n+16\n-13\n-17\n+5\n-18\n-1\n+12\n-9\n-18\n+25\n-3\n-8\n-15\n+9\n-19\n-5\n+18\n+5\n+3\n-1\n-12\n-12\n+8\n-21\n+3\n+1\n-32\n-8\n+2\n-3\n+27\n+12\n+38\n-17\n+42\n+27\n+10\n+5\n-1\n-1\n+10\n-17\n+11\n+18\n+2\n-19\n-11\n+18\n+7\n+23\n-5\n+21\n-20\n+2\n-6\n-20\n-7\n+16\n-8\n+41\n+15\n-3\n-18\n+16\n+3\n-2\n+16\n+8\n-1\n+3\n+17\n+21\n+6\n+19\n+8\n-14\n+8\n+4\n-1\n+10\n+7\n-9\n-20\n-9\n-60\n+40\n-14\n-19\n+38\n-52\n-13\n-203\n-31\n-15\n+1\n-9\n-12\n+17\n+10\n-9\n-5\n-21\n-5\n-6\n-11\n+19\n+10\n-11\n-16\n-37\n-22\n-13\n+5\n+19\n-14\n-7\n+13\n-8\n+26\n+16\n+19\n-7\n+28\n+16\n-2\n+18\n+27\n+13\n-150\n+5\n+22\n-6\n-22\n-14\n-136\n-91\n-7\n-14\n+31\n-278\n-73081\n+12\n+15\n-16\n+6\n-8\n-3\n+19\n-3\n+10\n+1\n+1\n+8\n+9\n+9\n-12\n+9\n-2\n+14\n+9\n-19\n+15\n+21\n+19\n+1\n+3\n+19\n-11\n+8\n+20\n-14\n+16\n+4\n+6\n-16\n-3\n-3\n-13\n-4\n+5\n+16\n-5\n-10\n+3\n-19\n-5\n+7\n-16\n-7\n+10\n+1\n+18\n+10\n-12\n-14\n-15\n-7\n+14\n+12\n+16\n+17\n+11\n+4\n+12\n+15\n-17\n+13\n-8\n+1\n-12\n-16\n-10\n-9\n-10\n+8\n-20\n+8\n-5\n-7\n-16\n+9\n-8\n-13\n-12\n-8\n+2\n+3\n+16\n-15\n+11\n+11\n+2\n-3\n-17\n-17\n-3\n-7\n-20\n+18\n-9\n+3\n-5\n-10\n-17\n+16\n-2\n-19\n-18\n-12\n-14\n-17\n-3\n+6\n+10\n+16\n+3\n+2\n-16\n-16\n-10\n-10\n+14\n-6\n-10\n+9\n+15\n+15\n+7\n-14\n-2\n+10\n-14\n-11\n-2\n+14\n+16\n+5\n+20\n+12\n-18\n+11\n+9\n+9\n-3\n-5\n+12\n+15\n+6\n-8\n+3\n+14\n+15\n+3\n-12\n-15\n+23\n-7\n+18\n-4\n+18\n+4\n+9\n+19\n-7\n+6\n+3\n-10\n+18\n-19\n-20\n-6\n-17\n-15\n+3\n+20\n-5\n-10\n-12\n-23\n+5\n-1\n-14\n-16\n-17\n-12\n+17\n-14\n+4\n-12\n+18\n+20\n-15\n-19\n-1\n-15\n+10\n-9\n+1\n+3\n-18\n+19\n-14\n-19\n+17\n-11\n-3\n-13\n+7\n+19\n-15\n+7\n-33\n-8\n+2\n+2\n-18\n-10\n+16\n-7\n+4\n-14\n+8\n+13\n-5\n+20\n-14\n+15\n-11\n-7\n-22\n-10\n+11\n+1\n+16\n-8\n+16\n-3\n+15\n+6\n-17\n+27\n+14\n+15\n+8\n+18\n+21\n-19\n-25\n+11\n+10\n+65\n-11\n-36\n-15\n-11\n-25\n+18\n+35\n+72\n+1\n+58\n+3\n+12\n+19\n-10\n+21\n+16\n+11\n+9\n+1\n+1\n+9\n-6\n+20\n-12\n-20\n+5\n+8\n-7\n+18\n+19\n-13\n-15\n-24\n+8\n-16\n+3\n+2\n-14\n-1\n-10\n-16\n-17\n+21\n+4\n+14\n-17\n+18\n-12\n+10\n+11\n+2\n-15\n-4\n-19\n+13\n+1\n-18\n-6\n-23\n-17\n+19\n+28\n+6\n+29\n+10\n+17\n-6\n+5\n-11\n+26\n+25\n-22\n+2\n-10\n-29\n+43\n+17\n+5\n+4\n+26\n+1\n+11\n+19\n-12\n+20\n+17\n-13\n+1\n+19\n+14\n-19\n-9\n-9\n+2\n+13\n+4\n-2\n+19\n+10\n-14\n+10\n-9\n+1\n+20\n+16\n+6\n-13\n+4\n+19\n+16\n-6\n-7\n-25\n-20\n+5\n+2\n-3\n-9\n-21\n-19\n+12\n-8\n-7\n+17\n-15\n-18\n-11\n+6\n+3\n+15\n-5\n-5\n-6\n-18\n-5\n+19\n-5\n+18\n-1\n+2\n-16\n-6\n-16\n+17\n-8\n+16\n+30\n+18\n+5\n-2\n+16\n+19\n-8\n+9\n-23\n+12\n+3\n+18\n+36\n+50\n-133\n+7\n+9\n-68\n+7\n+2\n-113\n-85\n-23\n-211\n-19\n-6\n-21\n-11\n-16\n+19\n+15\n-13\n-4\n+14\n+27\n-12\n-5\n-18\n-20\n-16\n-15\n-15\n-12\n+1\n+6\n+9\n-11\n+18\n+5\n-14\n-19\n-9\n+18\n+2\n-18\n-11\n+3\n+13\n-4\n-4\n+9\n-10\n-17\n-4\n+16\n+13\n+19\n-15\n+6\n-3\n+4\n+10\n+20\n+17\n-24\n-5\n+1\n+8\n+27\n+6\n+4\n+20\n+8\n-3\n+66\n-139\n-21\n+11\n-19\n+20\n-13\n-31\n+7\n-2\n+5\n-2\n+4\n-16\n+7\n+6\n-21\n-10\n-12\n+25\n-7\n-17\n-13\n+73906"
```

The last jump in memory size was from 7ish Gigs to 10ish Gigs.

Does Ruby need to continually allocate more memory when using `Array#===`?

History

#1 - 12/03/2018 03:43 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

`Array#===` is not `Array#member?`.

`freq` is an `Array` and `sum` is an `Integer`, `sofreq === sum` always returns false then cycle loops and `freq << sum` repeats forever.

#2 - 12/03/2018 03:45 AM - mame (Yusuke Endoh)

`Array#===` is irrelevant. Your code executes `freq << sum` which extends the array and consumes the memory infinitely, even if you use `Array#member?`. The difference is that `Array#===` is much faster than `Array#member?`. Thus, the memory consumption is also faster.

Note that `Array#===` is not equal to `Array#member?`. In fact, `Array` has no `===` method. It is `Object#===` which is equal to `Object#==` by default.

```
p [1,2,3] == 2      #=> false
p [1,2,3] == [1,2,3] #=> true
```

#3 - 12/03/2018 04:40 AM - danielpclark (Daniel P. Clark)

Thanks both of you! :-). I saw some bad advice on StackOverflow and new that case worked as follows:

```
case 2
when *[1,2,3]
  puts :winner
end
# => winner
```

So I had assumed it would work. My mistake.

Files

Screenshot from 2018-12-02 21-33-38.png
Screenshot from 2018-12-02 21-41-38.png

6.85 KB
7.23 KB

12/03/2018
12/03/2018

danielpclark (Daniel P. Clark)
danielpclark (Daniel P. Clark)