Ruby - Bug #21217

Integer.sqrt produces wrong results even on input <= 1e18

04/06/2025 05:00 AM - hjroh0315 (Matthew Roh)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	3.2.2	**Backport:**	3.2: REQUIRED, 3.3: DONE, 3.4: DONE

Description

Hello, I have been so far using Ruby in various online judge platforms that support it, while recently I discovered a very intriguing issue regarding Integer.sqrt.

Please refer to the two following verdicts on this recent problem that requires the usage of such a function:

- Python 3.11.4
- Ruby 3.2.2

The input constraints of the problem guarantees that n <= 1e18, and judging from the fact that implementing the same functionality using binary search gets a correct verdict, it is quite apparent that Integer.sqrt is producing wrong results even on input <= 1e18.

While I have not been able to reproduce this issue locally, it is not hard to imagine that the latest version of Ruby might still have the same issue considering there has been no similar bugfix request in the last 2~3 years.

I believe an easy way to fix this bug is to simply adjust the result to the correct one using a linear search (while(x*x<=n)x++; while(x*x>n)x--;) in the end.

Thank you for reading the bugfix request, and for your continued effort on maintaining the language.

Associated revisions

Revision 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b - 04/07/2025 02:08 AM - mame (Yusuke Endoh)

Fix Integer.sqrt to never exceed actual value

Integer.sqrt uses sqrt(3) from libm for small values.
This method must return a value less than or equal to the actual integer
square root, but libm's sqrt does not always guarantee that.

This change corrects that by decrementing the result if necessary.

Fixes [Bug #21217]

Revision 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b - 04/07/2025 02:08 AM - mame (Yusuke Endoh)

Fix Integer.sqrt to never exceed actual value

Integer.sqrt uses sqrt(3) from libm for small values.
This method must return a value less than or equal to the actual integer
square root, but libm's sqrt does not always guarantee that.

This change corrects that by decrementing the result if necessary.

Fixes [Bug #21217]

Revision 3a7b9ca9 - 04/07/2025 02:08 AM - mame (Yusuke Endoh)

Fix Integer.sqrt to never exceed actual value

Integer.sqrt uses sqrt(3) from libm for small values.
This method must return a value less than or equal to the actual integer
square root, but libm's sqrt does not always guarantee that.

This change corrects that by decrementing the result if necessary.

Fixes [Bug #21217]

Revision a67e9e41846cdadad9bb2d9e9d10223c52253898 - 04/07/2025 12:19 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b: [Backport #21217]

```
    Fix `Integer.sqrt` to never exceed actual value

    `Integer.sqrt` uses `sqrt(3)` from libm for small values.
    This method must return a value less than or equal to the actual integer
    square root, but libm's sqrt does not always guarantee that.

    This change corrects that by decrementing the result if necessary.

    Fixes [Bug #21217]
```

Revision a67e9e41846cdadad9bb2d9e9d10223c52253898 - 04/07/2025 12:19 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b: [Backport #21217]

```
    Fix `Integer.sqrt` to never exceed actual value

    `Integer.sqrt` uses `sqrt(3)` from libm for small values.
    This method must return a value less than or equal to the actual integer
    square root, but libm's sqrt does not always guarantee that.

    This change corrects that by decrementing the result if necessary.

    Fixes [Bug #21217]
```

Revision a67e9e41 - 04/07/2025 12:19 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b: [Backport #21217]

```
    Fix `Integer.sqrt` to never exceed actual value

    `Integer.sqrt` uses `sqrt(3)` from libm for small values.
    This method must return a value less than or equal to the actual integer
    square root, but libm's sqrt does not always guarantee that.

    This change corrects that by decrementing the result if necessary.

    Fixes [Bug #21217]
```

Revision 9abd48df705c37f69ec9fd954f84cbbbd65db624 - 04/14/2025 07:21 AM - k0kubun (Takashi Kokubun)

merge revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b: [Backport #21217]

```
    Fix `Integer.sqrt` to never exceed actual value

    `Integer.sqrt` uses `sqrt(3)` from libm for small values.
    This method must return a value less than or equal to the actual integer
    square root, but libm's sqrt does not always guarantee that.

    This change corrects that by decrementing the result if necessary.

    Fixes [Bug #21217]
```

Revision 9abd48df705c37f69ec9fd954f84cbbbd65db624 - 04/14/2025 07:21 AM - k0kubun (Takashi Kokubun)

merge revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b: [Backport #21217]

```
    Fix `Integer.sqrt` to never exceed actual value

    `Integer.sqrt` uses `sqrt(3)` from libm for small values.
    This method must return a value less than or equal to the actual integer
    square root, but libm's sqrt does not always guarantee that.

    This change corrects that by decrementing the result if necessary.

    Fixes [Bug #21217]
```

Revision 9abd48df - 04/14/2025 07:21 AM - k0kubun (Takashi Kokubun)

merge revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b: [Backport #21217]

```
    Fix `Integer.sqrt` to never exceed actual value
    `Integer.sqrt` uses `sqrt(3)` from libm for small values.
    This method must return a value less than or equal to the actual integer
    square root, but libm's sqrt does not always guarantee that.

    This change corrects that by decrementing the result if necessary.

    Fixes [Bug #21217]
```

History

#1 - 04/06/2025 09:13 AM - mame (Yusuke Endoh)

Thank you. I think it's most likely a Ruby bug, but I can't proceed until I identify the input that causes it.

I still believe Ruby is the cause, but I can't rule out a Python bug. (AtCoder's answer might have been generated with Python.)
Also, blindly applying linear correction is risky. If Ruby's result is far off due to the bug, the correction could take a huge amount of time.

#2 - 04/06/2025 09:20 AM - mame (Yusuke Endoh)

I was given a reproducible example!

https://x.com/tatyam_prime/status/1908810778276487443

```
irb(main):001> n = 4503599761588224
=> 4503599761588224
irb(main):002> Integer.sqrt(n) ** 2
=> 4503599761588225
```

#3 - 04/06/2025 09:52 AM - mame (Yusuke Endoh)

https://github.com/ruby/ruby/pull/13076

#4 - 04/06/2025 11:05 AM - hjroh0315 (Matthew Roh)

mame (Yusuke Endoh) wrote in #note-3:

> https://github.com/ruby/ruby/pull/13076

Thanks for the quick resolution!

#5 - 04/07/2025 01:23 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN to 3.2: REQUIRED, 3.3: REQUIRED, 3.4: REQUIRED

#6 - 04/07/2025 02:08 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

Applied in changeset git|3a7b9ca93b91dcc086b9ac8b9957e59268f9493b.

Fix Integer.sqrt to never exceed actual value

Integer.sqrt uses sqrt(3) from libm for small values.
This method must return a value less than or equal to the actual integer
square root, but libm's sqrt does not always guarantee that.

This change corrects that by decrementing the result if necessary.

Fixes [Bug #21217]

#7 - 04/07/2025 12:21 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 3.2: REQUIRED, 3.3: REQUIRED, 3.4: REQUIRED to 3.2: REQUIRED, 3.3: DONE, 3.4: REQUIRED

ruby_3_3 a67e9e41846cdadad9bb2d9e9d10223c52253898 merged revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b.

#8 - 04/14/2025 07:22 AM - k0kubun (Takashi Kokubun)

- Backport changed from 3.2: REQUIRED, 3.3: DONE, 3.4: REQUIRED to 3.2: REQUIRED, 3.3: DONE, 3.4: DONE

ruby_3_4 9abd48df705c37f69ec9fd954f84cbbbd65db624 merged revision(s) 3a7b9ca93b91dcc086b9ac8b9957e59268f9493b.