Previous Next

7. Reverse Integer

June 12, 2018 | 622.7K views

★★★★☆
Average Rating: 3.76 (316 votes)

Given a 32-bit signed integer, reverse digits of an integer.

Example 1:

```
Input: 123
Output: 321
```

Example 2:

```
Input: -123
Output: -321
```

Example 3:

```
Input: 120
Output: 21
```

Note:
Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: $[-2^{31}, \ 2^{31}-1]$. For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.

Solution

Approach 1: Pop and Push Digits & Check before Overflow

Intuition

We can build up the reverse integer one digit at a time. While doing so, we can check beforehand whether or not appending another digit would cause overflow.

Algorithm

Reversing an integer can be done similarly to reversing a string.

We want to repeatedly "pop" the last digit off of x and "push" it to the back of the rev. In the end, rev will be the reverse of the x.

To "pop" and "push" digits without the help of some auxiliary stack/array, we can use math.

```
//pop operation:
pop = x % 10;
x /= 10;

//push operation:
temp = rev * 10 + pop;
rev = temp;
```

However, this approach is dangerous, because the statement $\text{temp} = \text{rev} \cdot 10 + \text{pop}$ can cause overflow.

Luckily, it is easy to check beforehand whether or this statement would cause an overflow.

To explain, lets assume that rev is positive.

1. If $temp = \text{rev} \cdot 10 + \text{pop}$ causes overflow, then it must be that $\text{rev} \geq \frac{INTMAX}{10}$
2. If $\text{rev} > \frac{INTMAX}{10}$, then $temp = \text{rev} \cdot 10 + \text{pop}$ is guaranteed to overflow.
3. If $\text{rev} == \frac{INTMAX}{10}$, then $temp = \text{rev} \cdot 10 + \text{pop}$ will overflow if and only if $\text{pop} > 7$

Similar logic can be applied when rev is negative.

C++ Java Copy

```
class Solution {
public:
    int reverse(int x) {
        int rev = 0;
        while (x != 0) {
            int pop = x % 10;
            x /= 10;
            if (rev > INT_MAX/10 || (rev == INT_MAX / 10 && pop > 7)) return 0;
            if (rev < INT_MIN/10 || (rev == INT_MIN / 10 && pop < -8)) return 0;
            rev = rev * 10 + pop;
        }
        return rev;
    }
};
```

Complexity Analysis

- Time Complexity: $O(\log(x))$. There are roughly $\log_{10}(x)$ digits in x.
- Space Complexity: $O(1)$.

Rate this article: ★★★★☆

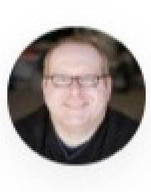
Previous Next

Comments: 623

Sort By

Type comment here... (Markdown is supported)

Preview Post



user4565t ★ 242 July 26, 2018 6:40 AM

I'm never sure whether it's a good thing or bad thing to show your knowledge of a language's standard library when whiteboarding exercises like this. If you're familiar enough with the Java standard lib to know a few of its nooks and crannies, then you can do this cleanly in just a few lines:

```
String reversed = new StringBuilder().append(Math.abs(x)).reverse().toString();
```

Read More

240 Share Reply

SHOW 27 REPLIES

Talisha ★ 430 August 15, 2018 7:31 PM

Can anyone explain the logic behind having condition pop > 7 and pop < -8?

427 Share Reply

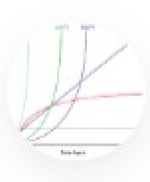
SHOW 45 REPLIES

ilgor ★ 197 January 2, 2019 2:11 AM

I don't get it why so many folks use solutions with long type and system accepts them? That trivializes the task, no? Problem description says: "Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range". Long wasn't a 32-bit signed last time I checked.

124 Share Reply

SHOW 8 REPLIES



winner_never_quit ★ 974 February 16, 2019 3:39 PM

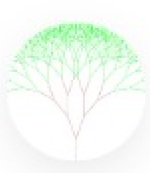
Java solution, same approach as article, but much cleaner:

```
    public int reverse(int x) {
        long res = 0;
        while (x != 0) {
```

Read More

280 Share Reply

SHOW 17 REPLIES



Dr_Sean ★ 535 December 21, 2018 12:27 PM

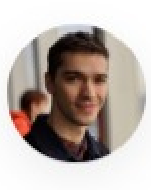
My Python code:

```
        if x >= 2**31-1 or x <= -2**31: return 0
        else:
            strg = str(x)
```

Read More

156 Share Reply

SHOW 11 REPLIES



scottdlindley ★ 74 November 4, 2018 3:48 AM

JS:

```
const reverse = x => {
    const limit = 2147483648;
    const k = x < 0 ? -1 : 1;
```

Read More

74 Share Reply

SHOW 12 REPLIES

Tavi3h ★ 94 November 5, 2018 12:42 PM

20ms / 99.97% in java:

```
    public int reverse(int x) {
        String ans = x < 0 ? new StringBuilder(String.valueOf(-x)).append("-").re
verse().toString()
```

Read More

41 Share Reply

SHOW 4 REPLIES

tprocroi ★ 75 April 5, 2019 12:17 AM

Python3 solution:

Uses a very fast string reverse slice notation -- convert `int` to `string`, reverse it, then back to `int`.

Read More

75 Share Reply

SHOW 17 REPLIES

califer ★ 39 October 2, 2018 11:40 AM

I think both two conditions are unneccessary
`|| (rev == INT_MAX / 10 && pop > 7)`
`|| (rev == INT_MAX / 10 && pop > 7)`
because when rev == INTMAX/10, pop then will be 0, 1, or 2 because the input is int.

38 Share Reply

SHOW 10 REPLIES

NocallerID ★ 77 August 2, 2018 2:30 PM

Easy Swift Implementation

```
  func reverse(_ x: Int) -> Int {
        var x = x
        var output: Int = 0
```

Read More

36 Share Reply

SHOW 2 REPLIES

‹ 1 2 3 4 5 6 ... 62 63 ›