9. Palindrome Number (/problems/palindrome-number/)

Aug. 31, 2017 | 41K views

Determine whether an integer is a palindrome. Do this without extra space.

click to show spoilers.

Some hints:
Could negative integers be palindromes? (ie, -1)

If you are thinking of converting the integer to string, note the restriction of using extra space.

You could also try reversing an integer. However, if you have solved the problem "Reverse Integer", you know that the reversed integer might overflow. How would you handle such case?

There is a more generic way of solving this problem.

Solution

Approach #1 Revert half of the number [Accepted]

Intuition

The first idea that comes to mind is to convert the number into string, and check if the string is a palindrome, but this would require extra non-constant space for creating the string which is not allowed by the problem description.

Second idea would be reverting the number itself, and then compare the number with original number, if they are the same, then the number is a palindrome. However, if the reversed number is larger than $\mathrm{int.MAX}$, we will hit integer overflow problem.

Following the thoughts based on the second idea, to avoid the overflow issue of the reverted number, what if we only revert half of the int number? After all, the reverse of the last half of the palindrome should be the same as the first half of the number, if the number is a palindrome.

For example, if the input is `1221`, if we can revert the last part of the number "12**21**" from "**21**" to "**12**", and compare it with the first half of the number "12", since 12 is the same as 12, we know that the number is a palindrome.

Let's see how we could translate this idea into an algorithm.

Algorithm

First of all we should take care of some edge cases. All negative numbers are not palindrome, for example: -123 is not a palindrome since the '-' does not equal to '3'. So we can return false for all negative numbers.

Now let's think about how to revert the last half of the number. For number `1221`, if we do `1221 % 10`, we get the last digit `1`, to get the second to the last digit, we need to remove the last digit from `1221`, we could do so by dividing it by 10, `1221 / 10 = 122`. Then we can get the last digit again by doing a modulus by 10,

`122 % 10 = 2`, and if we multiply the last digit by 10 and add the second last digit, `1 * 10 + 2 = 12`, it gives us the reverted number we want. Continuing this process would give us the reverted number with more digits.

Now the question is, how do we know that we've reached the half of the number?

Since we divided the number by 10, and multiplied the reversed number by 10, when the original number is less than the reversed number, it means we've processed half of the number digits.

C# Copy

```
public class Solution {
    public bool IsPalindrome(int x) {
        // Special cases:
        // As discussed above, when x < 0, x is not a palindrome.
        // Also if the last digit of the number is 0, in order to be a palindrome,
        // the first digit of the number also needs to be 0.
        // Only 0 satisfy this property.
        if(x < 0 || (x % 10 == 0 && x != 0)) {
            return false;
        }

        int revertedNumber = 0;
        while(x > revertedNumber) {
            revertedNumber = revertedNumber * 10 + x % 10;
            x /= 10;
        }

        // When the length is an odd number, we can get rid of the middle digit by revertedNumber/10
        // For example when the input is 12321, at the end of the while loop we get x = 12, revertedNumber = 123,
        // since the middle digit doesn't matter in palidrome(it will always equal to itself), we can simply get rid of it.
        return x == revertedNumber || x == revertedNumber/10;
    }
```

Complexity Analysis

- Time complexity : $O(log_{10}n)$. We divided the input by 10 for every iteration, so the time complexity is $O(log_{10}n)$
- Space complexity : $O(1)$.

Analysis written by: @ccwei (https://leetcode.com/ccwei)

Rate this article: (/ratings/107/233/?return=/articles/palindrome-number/) (/ratings/107/233/?return=/articles/palindrom

Previous (/articles/median-of-two-sorted-arrays/) Next (/articles/single-number/)

Join the conversation

Login to Reply

D (https://discuss.leetcode.com/user/dyywinner)

dyywinner commented 2 days ago

Python 3: we need use "int()" function to ensure that we have a integer argument.For example, "x = int(x/10)" and then "return (x == half or int(half / 10) == x)"

R (https://discuss.leetcode.com/user/rockbiker)

Rockbiker commented 6 days ago

If creating a String makes it more than O(1) spatial requirement than this is also exceed that space requirements. Note that you don't hold constant space, the consumption of space increase as the number gets larger.

To make this O(1) you simply need to allocate for the largest possible number - but this defeats the purpose.

V (https://discuss.leetcode.com/user/vkvk123vkvk321)

vkvk123vkvk321 commented last week

The downvote-to-upvote ratio is usually a good indicator.

This is a poorly setup problem. The core of it is good, but the explanation is bad. The "no space" thing is a lie since you are allocating space for revertedNumber.

P (https://discuss.leetcode.com/user/pouria_84-yahoo-com)

pouria_84-yahoo-com commented last week

@Mansi1997 (https://discuss.leetcode.com/uid/440571) Because we are not allowed to use extra space and creating a string violates this condition.

M (https://discuss.leetcode.com/user/mansi1997)

Mansi1997 commented 3 weeks ago

why can't we change it to string and then check??

G (https://discuss.leetcode.com/user/gamepla)

gamepla commented 3 weeks ago

return x == int(str(x)[::-1]) if x >= 0 else False

J (https://discuss.leetcode.com/user/jahuja)

jahuja commented 3 weeks ago

@andreyfedoseev (https://discuss.leetcode.com/uid/430749) Yaah, I understand it now. Also, I like the way of reversing just half digits as discussed in solution since it saves computation eventually.

A (https://discuss.leetcode.com/user/andreyfedoseev)

andreyfedoseev commented 3 weeks ago

@jahuja (https://discuss.leetcode.com/uid/230096) said in Palindrome Number (https://discuss.leetcode.com/post/244293):

> @andreyfedoseev (https://discuss.leetcode.com/uid/430749) so you mean if it 32768 then reversing will give 86723 so just catch an exception?

I don't think that 86723 will overflow, but generally yes, you just catch the integer overflow exception. If it's raised, then the original number is NOT a palindrome.

Here's why:

1. If the original number is a palindrome then the reversed number is the same as the original
2. If the original number does not overflow and it's a palindrome, then the reversed number does not overflow either because it's the same as the original
3. Consequently, if the reversed number overflows, then the original number is not a palindrome

J (https://discuss.leetcode.com/user/jahuja)

jahuja commented 3 weeks ago

@andreyfedoseev (https://discuss.leetcode.com/uid/430749) so you mean if it 32768(considering it as range limit) then reversing will give 86723 so just catch an exception?

A (https://discuss.leetcode.com/user/andreyfedoseev)

andreyfedoseev commented last month

> However, if the reversed number is larger than int.MAX, we will hit integer overflow problem.

It the reversed number is causing overflow it means that the original number is not a palindrome.

View original thread (https://discuss.leetcode.com/topic/101815)

Load more comments...