Previous Next

66. Plus One

March 7, 2020 | 27.4K views

★★★★★
Average Rating: 4.29 (14 votes)

Given a **non-empty** array of digits representing a non-negative integer, increment one to the integer.

The digits are stored such that the most significant digit is at the head of the list, and each element in the array contains a single digit.

You may assume the integer does not contain any leading zero, except the number 0 itself.

Example 1:

```
Input: [1,2,3]
Output: [1,2,4]
Explanation: The array represents the integer 123.
```

Example 2:

```
Input: [4,3,2,1]
Output: [4,3,2,2]
Explanation: The array represents the integer 4321.
```

Solution

Overview

"Plus One" is a subset of the problem set "Add Number", which shares the same solution pattern.

All these problems could be solved in linear time, and the question here is how to solve it without using the addition operation or how to solve it in constant space complexity.

The choice of algorithm should be based on the format of input. Here we list a few examples:

1. Integers

   Usually the addition operation is not allowed for such a case. Use Bit Manipulation Approach. Here is an example: Add Binary.

2. Strings

   Use bit by bit computation. Note, sometimes it might not be feasible to come up a solution with the constant space for languages with immutable strings, *e.g.* for Java and Python. Here is an example: Add Binary.

3. Linked Lists

   Sentinel Head + Schoolbook Addition with Carry. Here is an example: Plus One Linked List.

4. Arrays (also the current problem)

   Schoolbook addition with carry.

> Note that, a straightforward idea to convert everything into integers and then apply the addition could be risky, especially for the implementation in Java, due to the potential integer overflow issue.

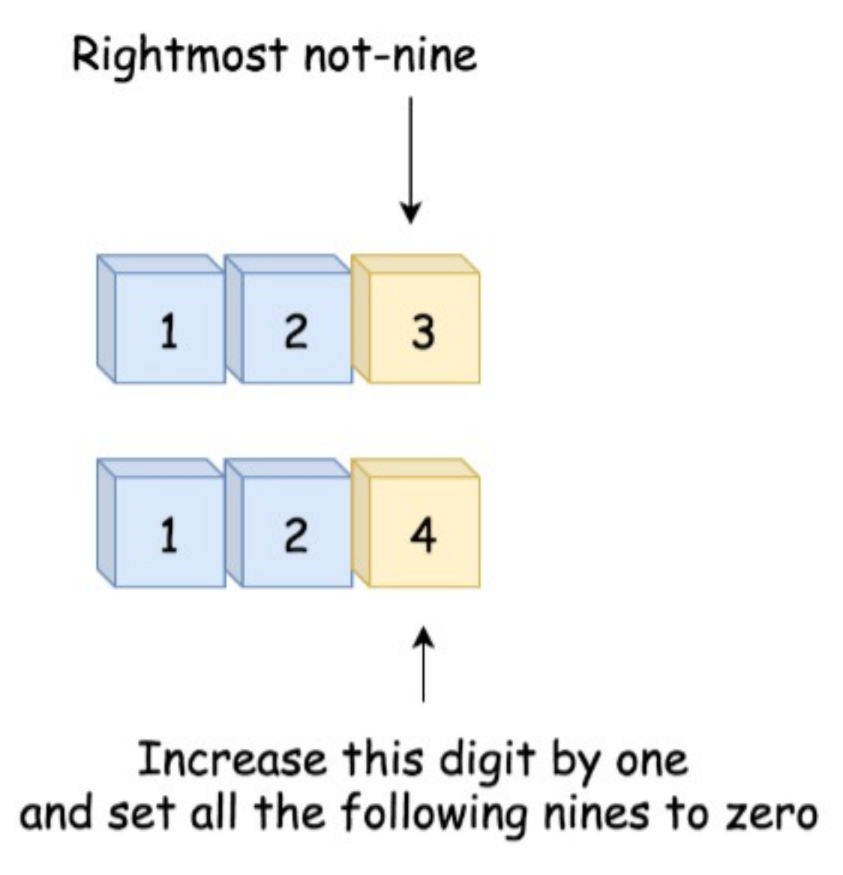
As one can imagine, once the array gets long, the result of conversion cannot fit into the data type of Integer, or Long, or even BigInteger.

Approach 1: Schoolbook Addition with Carry

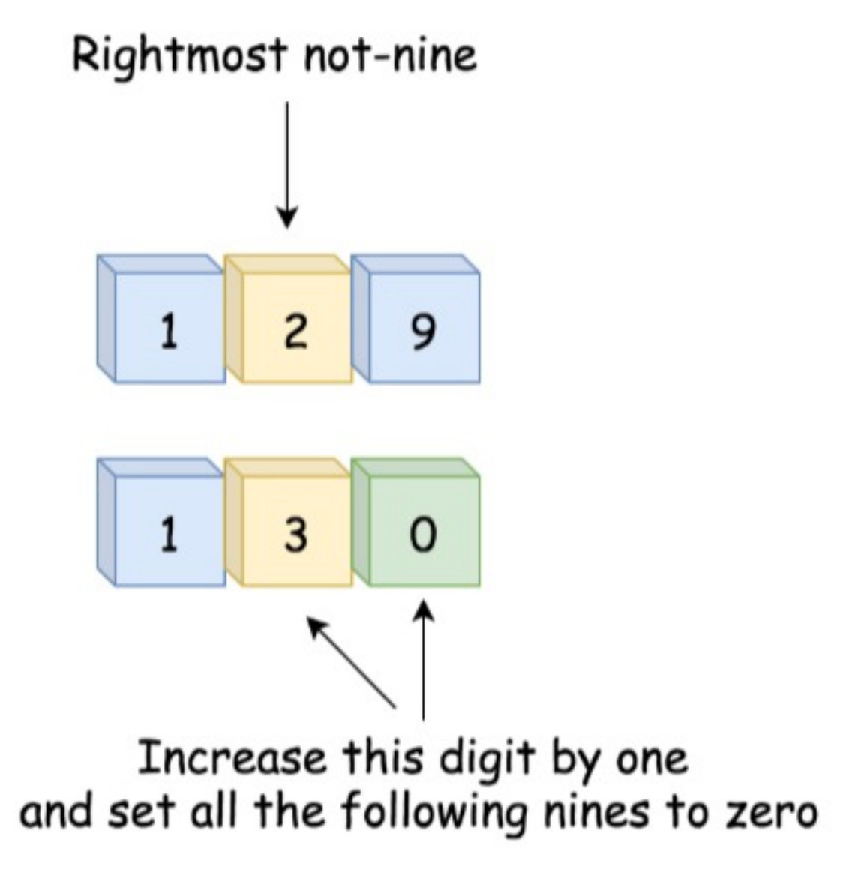
Intuition

Let us identify the rightmost digit which is not equal to nine and increase that digit by one. All the following consecutive digits of nine should be set to zero.

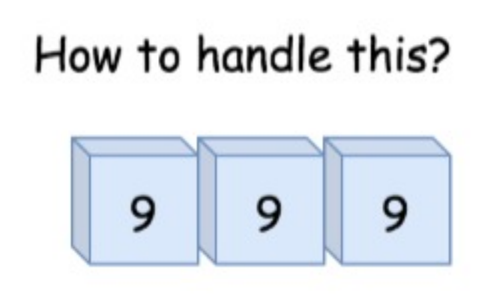
Here is the simplest use case which works fine.



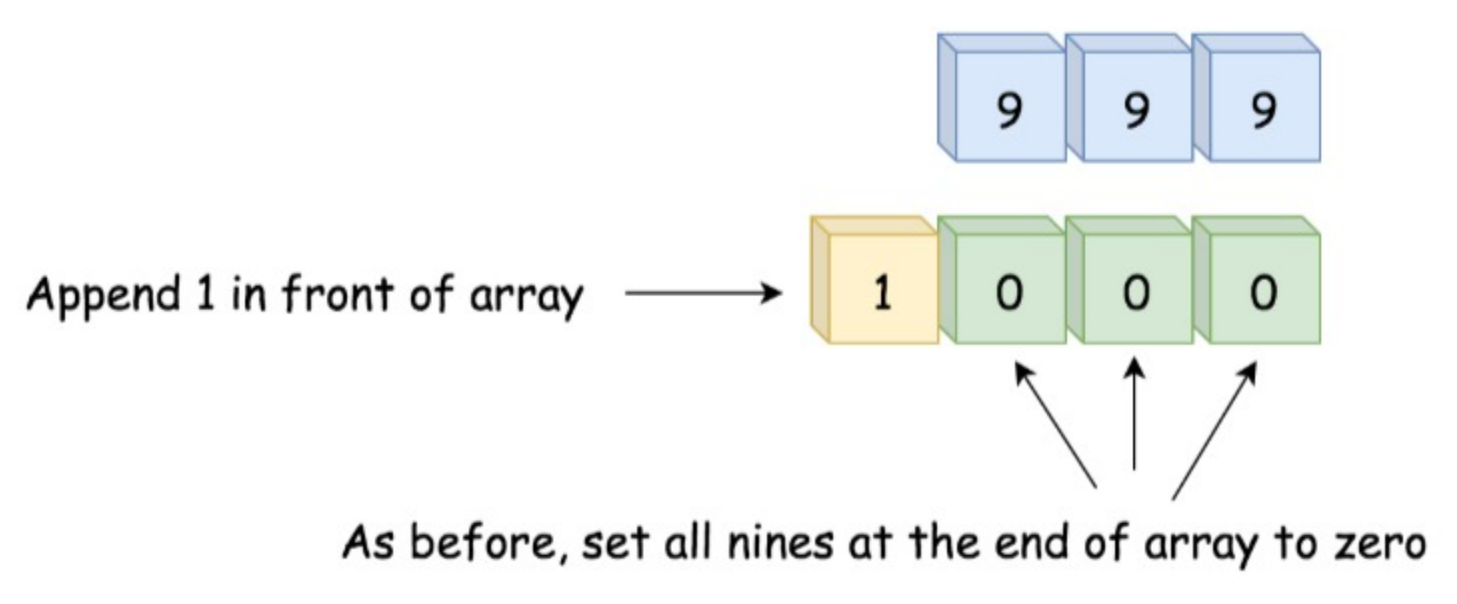
Here is a slightly complicated case which still passes.



And here is the case which breaks everything, because *all* the digits are nines.



In this case, we need to set all nines to zero and append 1 to the left side of the array.



Algorithm

- Move along the input array starting from the end of array.
- Set all the nines at the end of array to zero.
- If we meet a not-nine digit, we would increase it by one. The job is done - return `digits`.
- We're here because ***all*** the digits were equal to nine. Now they have all been set to zero. We then append the digit `1` in front of the other digits and return the result.

Implementation

Java Python Copy

```
class Solution:
    def plusOne(self, digits: List[int]) -> List[int]:
        n = len(digits)

        # move along the input array starting from the end
        for i in range(n):
            idx = n - 1 - i
            # set all the nines at the end of array to zeros
            if digits[idx] == 9:
                digits[idx] = 0
            # here we have the rightmost not-nine
            else:
                # increase this rightmost not-nine by 1
                digits[idx] += 1
                # and the job is done
                return digits

        # we're here because all the digits are nines
        return [1] + digits
```

Complexity Analysis

- Time complexity : $\mathcal{O}(N)$ since it's not more than one pass along the input list.
- Space complexity : $\mathcal{O}(1)$ when `digits` contains at least one not-nine digit, and $\mathcal{O}(N)$ otherwise.

Rate this article: ★★★★★

Previous Next

Comments: 7 Sort By

Type comment here... (Markdown is supported)

Preview Post

meowlicious99 ★ 476 March 14, 2020 11:46 PM

`digits = new int[n + 1];`

Why would space complexity be o(1) ?

11 Share Reply

SHOW 1 REPLY

roireshef ★ 39 April 14, 2020 6:25 PM

More readible python solution:

```
class Solution:
    def plusOne(self, digits: List[int]) -> List[int]:
        carry = 1
```

Read More

2 Share Reply

SHOW 1 REPLY

amolinasalazar ★ 6 June 14, 2020 12:59 AM

The description does not talk about if the change should be "in place" or not, but the thing is that this method has a different behaviour depending on the case:

All 9s: we create a new array in memory (heap), so we are returning a new different memory address. The pointer which the method was called will still hold the old memory address pointing to the all 0s

Read More

1 Share Reply

lidaivet ★ 80 April 15, 2020 7:51 AM

```
digits = new int[n + 1];
```

Is this a syntactic sugar that copies everything over?

Read More

1 Share Reply

SHOW 3 REPLIES

goldexp626 ★ 0 July 5, 2020 7:06 AM

Ruby cheated solution...

```
def plus_one(digits)
  (digits.join.to_i + 1).to_s.split('').map {|i| i.to_i}
end
```

Read More

0 Share Reply

SHOW 1 REPLY

wazza000 ★ 0 May 25, 2020 8:07 AM

My dum python solution:

```
class Solution:
    def plusOne(self, digits: List[int]) -> List[int]:
```

Read More

0 Share Reply

michealhahahaha ★ -2 May 9, 2020 1:01 AM

class Solution:
def plusOne(self, digits: List[int]) -> List[int]:
s=[str(i) for i in digits]
num=int(''.join(s))
numPlus=num+1

Read More

-2 Share Reply