66. Plus One

Easy 3795 4013 Add to List Share

You are given a **large integer** represented as an integer array `digits`, where each `digits[i]` is the i^{th} digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading `0`'s.

Increment the large integer by one and return *the resulting array of digits*.

Example 1:

```
Input: digits = [1,2,3]
Output: [1,2,4]
Explanation: The array represents the integer 123.
Incrementing by one gives 123 + 1 = 124.
Thus, the result should be [1,2,4].
```

Example 2:

```
Input: digits = [4,3,2,1]
Output: [4,3,2,2]
Explanation: The array represents the integer 4321.
Incrementing by one gives 4321 + 1 = 4322.
Thus, the result should be [4,3,2,2].
```

Example 3:

```
Input: digits = [9]
Output: [1,0]
Explanation: The array represents the integer 9.
Incrementing by one gives 9 + 1 = 10.
Thus, the result should be [1,0].
```

Constraints:

- 1 <= digits.length <= 100

Java Autocomplete

```java
class Solution {
    public int[] plusOne( int[] digits ) {

        int n = digits.length;

        for (int i = n - 1; i >= 0; i--) {

            if (digits[i] < 9) {
                digits[i]++;
                return digits;
            }

            digits[i] = 0;
        }

        int[] newNumber = new int[n + 1];
        newNumber[0] = 1;

        return newNumber;
    }
}
```

Your previous code was restored from your local storage. Reset to default

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input `[1,2,3]`

Output `[1,2,4]` Diff

Expected `[1,2,4]`