

[LeetCode] Plus One - 整数字符转换相加

原创

Eastmount

2015-09-09 05:19:20

4225

☆ 收藏

版权

分类专栏:

LeetCode

文章标签:

leetcode

数字整数转换相加



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、神...



Eastmount

¥9.90

订阅博主

题目概述:

Given a non-negative number represented as an array of digits, plus one to the number.

The digits are stored such that the most significant digit is at the head of the list.

题目解析:

给你一个int型数组存储一个非负整数，对整数加1后输出一个int型数组。注意几点：

- 1.可能存在进位操作，增加一位，如 $999+1=1000$ ；
- 2.数组存储如 $234=[2, 3, 4]$ ，它进行加1操作时从数组的高位(4)到低位(2)；
- 3.输出时也需要转置 $[0, 0, 0, 1]$ 转成1000；
- 4.C语言代码*returnSize是一维数组，注意赋值否则提示“超时异常”。

我的代码:

```
/**
 * Return an array of size *returnSize.
 * Note: The returned array must be malloced, assume caller calls free().
 * 899+1=900 存储时digits[3]=[8,9,9] 从高位到地位 result[3]=[0,0,9]需要转置
 * digitsSize数组长度 *returnSize为返回数组长度
 */
int* plusOne(int* digits, int digitsSize, int* returnSize) {
    // 初始时加1操作 后为进位数字0或1
    int add=1;
    int i,j=0;
    int temp;
    // 申请空间 初始化操作
    int *result=(int*)malloc(sizeof(int)*(digitsSize+1));
    memset(result, 0 , sizeof(int)*(digitsSize + 1));
    for(i=digitsSize-1; i>=0; i--) {
        result[j]=(digits[i]+add)%10;    // 个位数字
        add=(digits[i]+add)/10;          // 进位操作
        j++;
    }
}
```

```

//最后如果add==1表示位数加1 如99+1=100
if(add==1) {
    result[digitsSize]=1;
    *returnSize=digitsSize+1;    //注意它是一维数组
    //输出数组倒置
    for(i=0,j=digitsSize;i<j;i++,j--) {
        temp=result[i];
        result[i]=result[j];
        result[j]=temp;
    }
}
else {
    *returnSize=digitsSize;
    //输出数组倒置
    for(i=0,j=digitsSize-1;i<j;i++,j--) {
        temp=result[i];
        result[i]=result[j];
        result[j]=temp;
    }
}
return result;
}

```

推荐代码：

C语言代码 参考：<http://www.tonzoc.info/?p=688>

```

/**
 * Return an array of size *returnSize.
 * Note: The returned array must be malloced, assume caller calls free().
 */
void reverse(int* digits, int start, int end) {
    int temp;
    for (int i = start; i <= (start + end) >> 1; ++i) {
        temp = digits[i];
        digits[i] = digits[end + start - i];
        digits[end + start - i] = temp;
    }
}

int* plusOne(int* digits, int digitsSize, int* returnSize) {
    int num = 1;
    int* result = (int*)malloc(sizeof(int) * (digitsSize + 1));
    memset(result, 0, sizeof(int) * (digitsSize + 1));
    for (int i = digitsSize - 1; i >= 0; --i) {
        result[digitsSize - 1 - i] = (digits[i] + num) % 10;
    }
}

```

```

        num = (digits[i] + num) / 10;
    }
    if (num) {
        *returnSize = digitsSize + 1;
        result[digitsSize] = num;
        reverse(result, 0, digitsSize);
    } else {
        *returnSize = digitsSize;
        reverse(result, 0, digitsSize - 1);
    }
    return result;
}

```

C++代码

```

vector<int> plusOne(vector<int> &digits) {
    int carry=1, sum=0;
    vector<int> result(digits.size(),0);
    for(int i=digits.size()-1;i>=0;i--){
        sum = carry+digits[i];
        carry = sum/10;
        result[i] = sum%10;
    }
    if(carry>0){ //进位
        result.insert(result.begin(),carry);
    }
    return result;
}

```

同类题目：

二进制字符串加法 <https://leetcode.com/problems/add-binary/>

PS：需要注意转换方法 $((a[i] - '0') + (b[j] - '0') + add) \% 2 + '0'$ 当前结果和进位数字 $add = ((a[i] - '0') + (b[j] - '0') + add) / 2$ ；同时需要注意字符串倒置的方法和对齐判断即可。

(By:Eastmount 2015-9-9 凌晨2点 <http://blog.csdn.net/eastmount/>)