

445. Add Two Numbers II

[link](#)

这里尝试用int 然后reverse 最后出现超过int的异常, 还是得用stack做.

```
public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
    Stack<Integer> s1 = new Stack();
    Stack<Integer> s2 = new Stack();
    ListNode c1 = l1, c2 = l2, res = null, node = null;
    int first = 0, second = 0, carry = 0, digit = 0;
    while(c1 != null){
        s1.push(c1.val);
        c1 = c1.next;
    }
    while(c2 != null){
        s2.push(c2.val);
        c2 = c2.next;
    }
    while(!s1.isEmpty() || !s2.isEmpty()){
        if(s1.isEmpty()){
            second = s2.pop();
            digit = (second + carry) % 10;
            carry = (second + carry) > 9 ? 1 : 0;
        }else if(s2.isEmpty()){
            first = s1.pop();
            digit = (first + carry) % 10;
            carry = (first + carry) > 9 ? 1 : 0;
        }else{
            first = s1.pop();
            second = s2.pop();
            digit = (first + second + carry) % 10;
            carry = (first + second + carry) > 9 ? 1 : 0;
        }
        node = new ListNode(digit);
        node.next = res;
        res = node;
    }
    if(carry == 1){
        node = new ListNode(1);
        node.next = res;
        res = node;
    }
    return res;
}
```

更好的是翻转结果. 先存着超过10的部分, 然后第二次翻转的时候carry加进去

```
ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
```

```

int n1 = 0, n2 = 0, carry = 0;
ListNode *curr1 = l1, *curr2 = l2, *res = NULL;
while( curr1 ){ curr1=curr1->next; n1++; }
while( curr2 ){ curr2=curr2->next; n2++; }
curr1 = l1; curr2 = l2;
while( n1 > 0 && n2 > 0 ){
    int sum = 0;
    if( n1 >= n2 ){
        sum += curr1->val;
        curr1=curr1->next;
        n1--;
    }
    if( n2 > n1 ){
        sum += curr2->val;
        curr2=curr2->next;
        n2--;
    }
    res = addToFront( sum, res );
}
curr1 = res; res = NULL;
while( curr1 ){
    curr1->val += carry; carry = curr1->val/10;
    res = addToFront( curr1->val%10, res );
    curr2 = curr1;
    curr1 = curr1->next;
    delete curr2;
}
if( carry ) res = addToFront( 1, res );
return res;
}
ListNode* addToFront( int val, ListNode* head ){
    ListNode* temp = new ListNode(val);
    temp->next = head;
    return temp;
}

```

自己后来又写了一遍

```

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
    int len1 = 0, len2 = 0;
    ListNode c1 = l1, c2 = l2;
    while(c1 != null){
        len1++;
        c1 = c1.next;
    }

    while(c2 != null){
        len2++;
        c2 = c2.next;
    }
    int diff = Math.abs(len1 - len2);

    if(len1 > len2){
        c1 = l1;
        c2 = l2;
    }else{
        c1 = l2;
        c2 = l1;
    }
}

```

```

ListNode dummy = new ListNode(1);
ListNode current = dummy;
for(int i = 0; i < diff; ++i){
    current.next = new ListNode(c1.val);
    current = current.next;
    c1 = c1.next;
}
while(c1 != null){
    current.next = new ListNode(c1.val + c2.val);
    current = current.next;
    c1 = c1.next;
    c2 = c2.next;
}
ListNode head = null;
current = dummy.next;
ListNode next = null;
while(current != null){
    next = current.next;
    current.next = head;
    head = current;
    current = next;
}
while(head != null){
    if(head.val >= 10 && head.next != null){
        head.val -= 10;
        head.next.val++;
    }
    next = head.next;
    head.next = current;
    current = head;
    head = next;
}
if(current.val >= 10){
    current.val -= 10;
    dummy.next = current;
    return dummy;
}
return current;
}

```