

第27题目——求 $1+2+\dots+n$

求 $1+2+\dots+n$ ，要求不能使用乘除法、for、while、if、else、switch、case等关键字及条件判断语句（A?B:C）。

示例 1:

输入：n = 3
输出：6

示例 2:

输入：n = 9
输出：45

限制:

- $1 \leq n \leq 10000$

递归法（1）

这道题乍一看非常简单，但题目限制了我们的关键字。

倘若这道题没有限制，那么我第一时间想到的就是：

```
int sumNums(int n){  
    return n==0 ? 0 : n+sumNums(n-1);  
}
```

但现在我们既没有条件判断关键字，也没有三目运算符，怎么办呢？

稍微思考一下，我们依旧可以使用递归解决

以下是leetcode官方的解题思路：

以逻辑运算符 `&&` 为例，对于 `A && B` 这个表达式，如果 `A` 表达式返回 `\textit{False}`，那么 `A && B` 已经确定为 `\textit{False}`，此时不会去执行表达式 `B`。同理，对于逻辑运算符 `||`，对于 `A || B` 这个表达式，如果 `A` 表达式返回 `\textit{True}`，那么 `A || B` 已经确定为 `\textit{True}`，此时不会去执行表达式 `B`。

利用这一特性，我们可以将判断是否为递归的出口看作 `A && B` 表达式中的 `A` 部分，递归的主体函数看作 `B` 部分。如果不是递归出口，则返回 `\textit{True}`，并继续执行表达式 `B` 的部分，否则递归结束。当然，你也可以用逻辑运算符 `||` 给出类似的实现，这里我们只提供结合逻辑运算符 `&&` 的递归实现。

```
int sumNums(int n){  
    n>0 && (n+=sumNums(n-1));  
    return n;  
}
```

我们使用 && 则程序会得到两个前提，第一， $n > 0$ ，也就是说， n 一定为真。第二， $n += \text{sumNums}(n-1)$

则该条递归语句就可以累加所有的项，然后返回 n 。