

report

week 3

程雨歌 12307110079

2016 年 3 月 19 日

1 isValid.c

1.1 函数说明

输入 指向一个字符串的指针

输出 一个 Bool 值：该字符串中三种括号‘(’、’)’，‘[’、’]’和‘{’、’}’是否符合规则

实现方法 以数组实现栈。依次遍历字符串，
遇到左括号则压栈对应数字‘(’：1，‘[’：2，‘{’：3)；
遇到右括号则检查栈顶是否为对应的左括号：
若对应则弹栈；
若不对应则返回 false；
结束后若栈空则返回 true，若不为空则返回 false。

1.2 复杂度

时间复杂度 $\mathcal{O}(n)$

空间复杂度 $\mathcal{O}(1)$

1.3 边界情况

当输入为空字符串时没有进行压栈，最后判断栈为空，输出 true；

当字符串长度超过设定的 MAXLEN 时会出现栈溢出。

若采用头插法的链表来实现栈，可以将空间复杂度从 $\mathcal{O}(1)$ “降”为 $\mathcal{O}(n)$ ，此时栈溢出取决于运行电脑的内存，而不是设定的 MAXLEN。但由于无法在函数内部声明链表结构体，只得用更为简单的数组来实现栈。

1.4 程序运行结果

...

Score:12 / 12

2 getIntersectionNode.c

2.1 函数说明

输入 两个链表的头节点

输出 两个链表合并的节点

实现方法 先求出两个链表的长度，然后将两个链表后端对齐，从短链表的头节点和长链表的对应节点开始比较，直到出现相同节点，即为合并点。

注意只需对比节点本身（地址）即可，不同节点可能具有相同的值。

2.2 复杂度

时间复杂度 $\mathcal{O}(n)$

空间复杂度 $\mathcal{O}(1)$

2.3 边界情况

存在空链表时，对比过程不进行，直接进入最后返回 NULL。

若两链表无合并，则知道对比结束也不会有匹配，返回 NULL。

2.4 程序运行结果

...

Correct answer.

Start testing with 10000 random cases...

Your code runs well!