

重 庆 交 通 大 学

《 算 法 与 数 据 结 构 》 课 程

实 验 报 告

班 级： 计算机专业 19 级 曙光班

姓 名 学 号： 周迎川 631907060434

实验项目名称： 栈实验

实验项目性质： 验证性实验

实验所属课程： 数据结构 A

实验室(中心)： B01 409

指 导 教 师： 鲁云平

实验完成时间： 2020 年 10 月 22 日

教师评阅意见：

签名： 年 月 日

实验成绩：

一、实验目的

实现栈的存储结构, 通过实验深入理解栈的操作特点.

二、实验内容及要求

实验内容：

1. 实现栈的存储结构及相关操作：进栈、出栈、取栈顶元素等
2. 使用该栈完成一个字符串的逆序输出
3. 使用该栈完成表达式的括号是否匹配？
4. 对算术表达式求值(选作)

- 要求：
1. 栈的存储结构自定, 并使用栈完成一个字符串的逆序输出
 2. 使用该栈完成表达式的括号是否匹配？
 3. 对算术表达式求值(选作)
 4. 独立完成实验内容
 5. 自行实现栈的存储结构与相关操作，不得使用 STL(标准模板库) 现成

代码：

6. 数据元素的类型自定；
- 7 . 编程语言：C++

三、系统分析

(1) 数据方面：

根据实验要求，需要将字符串倒置，因此可以定义为 `char` 类型，判断括号匹配问题，括号可以用 `char` 存储。本实验采用链式栈。

(2) 功能方面：

栈必须要压栈，出栈，获取栈顶元素，在本实验中使用了链式栈不会栈溢出，也不需要判满操作，只需要判空。

四、系统设计

(1) 设计的主要思路

说明整体设计思路

根据题目要求，将字符串倒置，优先考虑栈数据结构，实验用模板类，然后再用链式栈的结构。

(2) 数据结构的设计

数据结构设计思路

整个系统主要采用链式栈的数据结构，首先定义一个 Stack 类，然后再定义一个链式栈的结点类 ListNode，再定义一个链式栈类 ListStack，从 Stack 类继承。

```
template <typename T>
class Stack
{
public:
    Stack()//构造函数
    {}
    virtual bool Push(T &x)=0;//入栈
    virtual bool Pop(T &x)=0;//出栈
    virtual bool getTop(T &x) const=0;//读取栈顶元素
    virtual bool IsEmpty() const=0;//判空
    virtual int getSize() const = 0;//计算栈的长度
};

template <typename T>
struct ListNode//链表结点类
{
    T data;
    ListNode<T> *next;
    ListNode(const T &x, ListNode<T> *ptr=nullptr)//构造链表结点
    {
        data=x; next=ptr;
    }
};

template <typename T>
class ListStack: public Stack<T>
{
private:
    ListNode<T> *top;
public:
    ListStack()//构造函数，将链式栈的 top 指针赋值为空
    {
        top=nullptr;
    }
    ~ListStack()//析构函数，将链式栈置为空
```

```

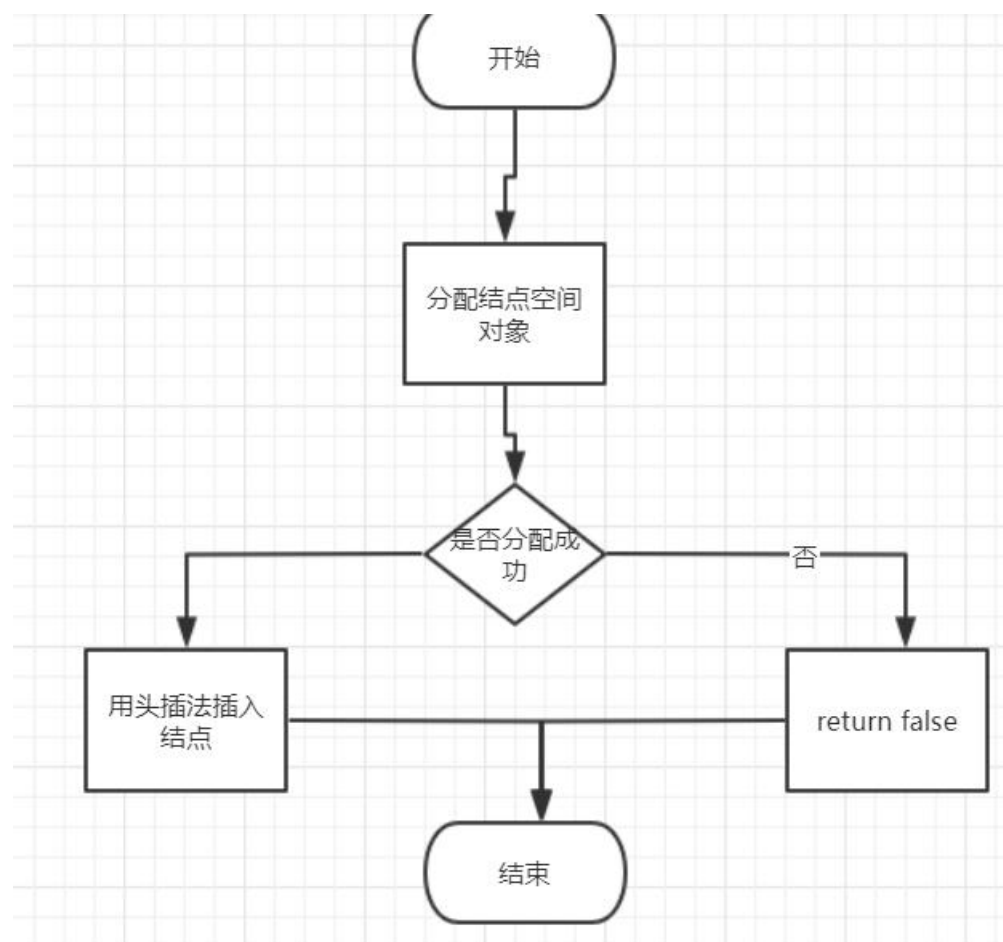
{
    makeEmpty();
}
bool Push(T &x); //入栈
bool Pop(T &x); //出栈
bool getTop(T &x) const; //读取栈顶元素
void makeEmpty(); //置空
bool IsEmpty() const; //判空
int getSize() const; //获取链式栈的长度
//friend ostream& operator<<(ostream &out, ListStack<T> &s);
};

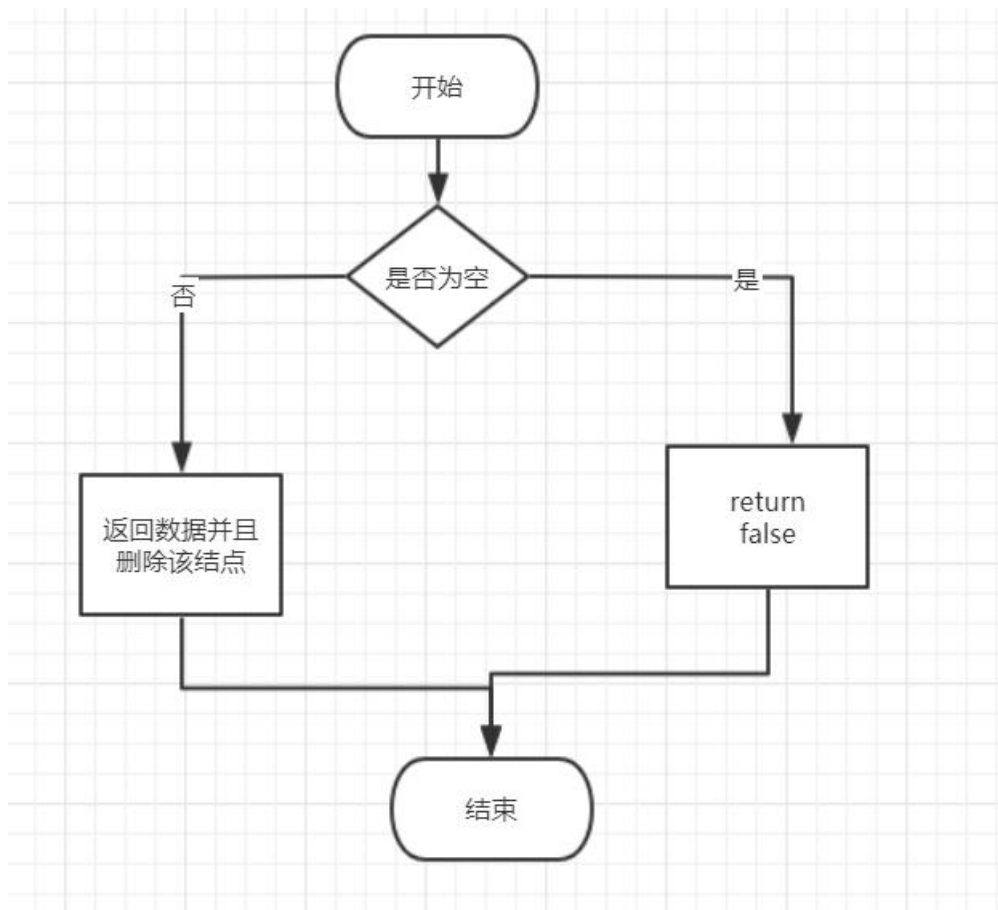
```

(3) 基本操作的设计

基本操作的抽象描述，关键算法的设计思路和算法流程图。

基本操作的抽象描述一般为操作名，初始条件，操作结构，参数说明等。





五、编程环境与实验步骤

(1) 编程环境

主要是操作系统、编程工具软件

主要操作系统：Windows 10

编程工具：Qt

(2) 实验步骤

只说明程序相关的各种文件创建步骤及文件的作用，不需说明文件的具体内容。

stack.h 栈头文件

stack.cpp 栈功能的实现文件

main.cpp 测试功能文件

(3) 编译参数

若有特殊的编译参数设置，需说明详细步骤。

若无特殊的编译参数设置，则只需简单说明操作步骤。

六、实现代码

主要功能的实现代码

```
#include "stack.h"
#include <cstdlib>
#include <iostream>
using namespace std;
template <typename T>
bool ListStack<T>::Push(T &x)
{
    //入栈就是链表的前插，链表的头指针就是 top 指针
    ListNode<T> *newnode=new ListNode<T>(x);
    if(newnode==nullptr)
        cerr<<"分配储存错误！！！"<<endl;
    newnode->next=top;
    top=newnode;
    return true;
}
template <typename T>
bool ListStack<T>::Pop(T &x)
{
    //如果链式栈为空，则不能 Pop 返回 false
    if(IsEmpty())
        return false;
    ListNode<T> *now=top;
    top=now->next;
    x=now->data;
    delete now;
    return true;
}
template <typename T>
bool ListStack<T>::getTop(T &x) const
{
    //如果链式栈为空，返回 false
    if(IsEmpty())
        return false;
    x=top->data;
    return false;
}
template <typename T>
void ListStack<T>::makeEmpty()
{
    //如果说现在的指针不是指向 nullptr，则还可以继续删除
    ListNode<T> *now;
    while(top!=nullptr)
    {
```

```

        now=top;
        top=top->next;
        delete now;
    }
}
template <typename T>
bool ListStack<T>::IsEmpty() const
{
    //如果说链式栈的 top 指针指向 nullptr 则为空
    if(top==nullptr)
        return true;
    else
        return false;
}
template <typename T>
int ListStack<T>::getSize() const
{
    ListNode<T> *now=top;
    int account=0;
    while(now!=nullptr)
    {
        account++;
        now=now->next;
    }
    return account;
}
//template <typename T>
//ostream& operator<<(ostream &out, ListStack<T> &s)
//{
//    out<<s.data<<endl;
//}

```

七、测试结果与说明

至少完成功能测试，使用测试数据测试相关功能是否符合设计要求。


```
输入字符串
abcdefg
栈顶元素g
栈的长度7
g
f
e
d
c
b
a
输入括号
((((((())))
有与第7个括号匹配的括号!!!
有与第8个括号匹配的括号!!!
有与第9个括号匹配的括号!!!
有与第10个括号匹配的括号!!!
还有“(”没有匹配
_
```

八、实验分析

(1) 算法的性能分析

主要针对增加、删除、搜索等算法。

入栈： $O(1)$

出栈： $O(1)$

(2) 数据结构的分析

通过性能分析总结此种存储结构的优缺点，并说明其适用场景。

链式栈主要是先入栈的后出栈，所以此数据结构使用于先入后出的场景，比如说将字符串倒置和函数的递归调用。

优点：可以将某些功能有序进行，比如说函数的递归调用，不会混乱，就是用的递归。

缺点：不能随机访问，只能从栈丁开始访问，只要栈丁元素没有出栈的话后面的元素是不能出栈的。

九、实验总结

主要针对本实验的分析、设计、实现、测试等环节进行总结，包含收获与不足，此部分的阐述应较为详细。

此次实验验证了栈的规律于正确性，了解了栈的应用之一-----括号匹配，也让我们熟悉了栈的实现，从理论到时间的过程有些许坎坷，我们不能停留在理论的层面，必须要动手去实践。

实现链式栈的时候，我开始压栈的时候出现了断链的情况，因为我对链式栈的不熟悉，导致我对压栈的实现出现了问题，通过百度解决了此问题。在验证括号是否匹配的时候，开始有情况没有考虑完整，导致输出结果不是想要的，后面查看了数据结构书上的代码。

附录

参考文献：

- 1.
- 2.
- 3.