

# 计算机学院 高级语言程序设计 课程实验报告

实验题目：通用排序算法、模板特化、偏特化、元编程		学号：202200400053
日期：2024-05-30	班级：2202	姓名：王宇涵
Email： <a href="mailto:1941497679@qq.com">1941497679@qq.com</a>		
<b>实验目的：</b> 1. 实现通用排序算法编程 2. 会使用模板特化、偏特化、元编程		
<b>实验软件和硬件环境：</b> 软件环境：VSCODE + DEV-C++ 硬件环境：Legion Y7000P		
<b>实验步骤与内容：</b> 1. 练习习题 9.4，利用 DNode.h，编写使用双向链表的程序，其中实现左插入、右插入，删除，显示等链表的功能。 实现双向链表类定义如下 <pre>template &lt;class T&gt; class DoublyLinkedList { public:     DNode&lt;T&gt; *head;     DoublyLinkedList();     ~DoublyLinkedList();      void insertRight(const T&amp; item);     void insertLeft(const T&amp; item);     void deleteNode(DNode&lt;T&gt; *node);     void display() const; };</pre>		
测试程序如下		

```

int main()
{
    DoublyLinkedList<int> list;

    std::cout << "Inserting 10, 20, 30 to the right:" << std::endl;
    list.insertRight(10);
    list.insertRight(20);
    list.insertRight(30);
    list.display();

    std::cout << "Inserting 5 to the left:" << std::endl;
    list.insertLeft(5);
    list.display();

    std::cout << "Deleting the first node:" << std::endl;
    DNode<int> *firstNode = list.head->nextNodeRight();
    list.deleteNode(firstNode);
    list.display();

    std::cout << "Deleting the last node:" << std::endl;
    DNode<int> *lastNode = list.head->nextNodeLeft();
    list.deleteNode(lastNode);
    list.display();

    return 0;
}

```

输出结果:

```

Inserting 10, 20, 30 to the right:
30 20 10
Inserting 5 to the left:
30 20 10 5
Deleting the first node:
20 10 5
Deleting the last node:
20 10

```

成功完成功能

2. 通用排序算法练习，并讨论以下排序算法的特点

- (1) 练习习题 9-10，插入排序。
- (2) 练习习题 9-12，选择排序。
- (3) 练习习题 9-14，交换排序（起泡排序）。

(1)插入排序

排序前的数据:

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

开始排序...

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 2 3 5 7 9 11 13 15 17 19 4 6 8 10 12 14 16 18 20

1 2 3 4 5 7 9 11 13 15 17 19 6 8 10 12 14 16 18 20

1 2 3 4 5 6 7 9 11 13 15 17 19 8 10 12 14 16 18 20

1 2 3 4 5 6 7 8 9 11 13 15 17 19 10 12 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 13 15 17 19 12 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 15 17 19 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 19 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 19 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

排序后的数据:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

## (2)选择排序

F:\BaiduSyncDisk\CLASSES\c++\exp\exp11\code\output> g++ ex9-12.exe

排序前的数据:

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

开始排序...

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

1 2 5 7 9 11 13 15 17 19 3 4 6 8 10 12 14 16 18 20

1 2 3 7 9 11 13 15 17 19 5 4 6 8 10 12 14 16 18 20

1 2 3 4 9 11 13 15 17 19 5 7 6 8 10 12 14 16 18 20

1 2 3 4 5 11 13 15 17 19 9 7 6 8 10 12 14 16 18 20

1 2 3 4 5 6 13 15 17 19 9 7 11 8 10 12 14 16 18 20

1 2 3 4 5 6 7 15 17 19 9 13 11 8 10 12 14 16 18 20

1 2 3 4 5 6 7 8 17 19 9 13 11 15 10 12 14 16 18 20

1 2 3 4 5 6 7 8 9 19 17 13 11 15 10 12 14 16 18 20

1 2 3 4 5 6 7 8 9 10 17 13 11 15 19 12 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 13 17 15 19 12 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 17 15 19 13 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 15 19 17 14 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 19 17 15 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 19 16 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 19 17 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 19 18 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

排序后的数据:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

## (3)交换排序

排序前的数据：

1 3 5 7 9 11 13 15 17 19 2 4 6 8 10 12 14 16 18 20

开始排序...

1 3 5 7 9 11 13 15 17 2 4 6 8 10 12 14 16 18 19 20

1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16 17 18 19 20

1 3 5 7 9 11 13 2 4 6 8 10 12 14 15 16 17 18 19 20

1 3 5 7 9 11 2 4 6 8 10 12 13 14 15 16 17 18 19 20

1 3 5 7 9 2 4 6 8 10 11 12 13 14 15 16 17 18 19 20

1 3 5 7 2 4 6 8 9 10 11 12 13 14 15 16 17 18 19 20

1 3 5 2 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 3 2 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

排序后的数据：

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

三种排序的特点：

选择排序：每次从待排序的数据中选择最小（或最大）的元素，放到已排序序列的末尾，直到全部元素排序完成。

冒泡排序：相邻元素比较，若顺序错误则交换，一次遍历可确保最大（或最小）元素移动到正确位置。

插入排序：将待排序的元素逐个插入已排序序列的合适位置，直到全部元素排序完成。

3. 练习习题 9-19，折半查找。该算法的前提是？

练习

输入想查找的数字(1~20)：5

数据为：

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

5是第5个数字

PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output>

输入想查找的数字(1~20)：0

数据为：

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

没有找到数字0

PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output>

算法的前提：数组是有序的。

4. 模板特化：

(1)练习习题 9-22，模板特化，函数模板重载。

练习：

PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> .\ex9-22\_1.exe

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> .\ex9-22\_2.exe

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

(2)实践第9章 PPT,对例 9-8,Stack 栈类模板进行偏特化,使得 teststack.cpp 可以测试运行,Stack.h 中还需补充哪些成员函数实现？

补充函数

```
template <int SIZE>
void Stack<bool, SIZE>::clear() {
    top = -1;
}
```

测试结果

```
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> & .\ex9-23_1.exe
size:16 top:3
size:8 top:1
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output>
```

## 5. 模板元编程:

(1)参照第 9 章 PPT, P100 例编写 Power.h,使得 testPower.cpp 可以测试运行。

测试结果:

```
16
64
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output>
```

(2)练习第 9 章习题 9-23, 模板元编程。

测试结果:

求排列数

```
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> & .\ex9-23_1.exe
90
```

求最大公约数

```
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> & .\ex9-23_2.exe
6
```

## 6. 可变参数个数函数模板:

(1) 第 9 章 PPT, P102 例,实践将"递归终止版本"写成 void print() {}

(2) 实践第 9 章 PPT, P104 (非递归版本) 例, 需加编译选项 -std=c++17 或-std=gnu++17

(1)练习

```
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> & .\9_5_4.exe
3
4.5
hello world!
```

实践

```
// 递归终止版本
void print() { }
```

输出

```
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> & .\9_5_4.exe
3
4.5
hello world!
```

(2) 实践结果

```
PS D:\BaiduSyncdisk\CLASSES\C++\exp\exp11\code\output> & .\'9_5_4_noRecursive.exe'  
2  
4.5  
hello world!
```

此时 2 为参数个数, 后者为参数

## 7. 练习课本综合实例 9.4, 个人银行账户管理程序, 实现例 9-16。

练习结果:

```
(a)add account (d)deposit (w)withdraw (s)show (c)change day (n)next month (e)exit  
2008-11-1      Total: 0      command> a s S12345 0.03  
2008-11-1      #S12345 created  
2008-11-1      Total: 0      command> a c C67890 1000 0.02 5  
2008-11-1      #C67890 created  
2008-11-1      Total: 0      command> d 0 500 Deposit to savings  
2008-11-1      #S12345 500    500      Deposit to savings  
2008-11-1      Total: 500    command> d 1 200 Deposit to credit  
2008-11-1      #C67890 200    200      Deposit to credit  
2008-11-1      Total: 700    command> w 0 100 Withdraw from savings  
2008-11-1      #S12345 -100   400      Withdraw from savings  
2008-11-1      Total: 600    command> w 1 50 Withdraw from credit  
2008-11-1      #C67890 -50    150      Withdraw from credit  
2008-11-1      Total: 550    command> s  
[0] S12345      Balance: 400  
[1] C67890      Balance: 150    Available credit:1000  
2008-11-1      Total: 550    command> c 15  
2008-11-15     Total: 550    command> n  
2008-12-1      Total: 550    command> s  
[0] S12345      Balance: 400  
[1] C67890      Balance: 150    Available credit:1000  
2008-12-1      Total: 550    command> e
```

### 结论分析与体会:

在本次实验中, 我深入理解并实现了通用排序算法, 包括插入排序、选择排序和冒泡排序, 并分析了它们的特点和适用场景。

通过编写和调试使用双向链表的程序, 我掌握了链表的基本操作如左插入、右插入和删除等。

进一步, 我学习了模板特化和偏特化, 特别是通过实现偏特化的 `Stack` 类模板, 增强了对模板编程的理解。

此外, 通过实践模板元编程, 我理解了编译期计算的强大功能。

最后, 尝试实现可变参数模板函数, 加深了对 C++17 新特性的认识。

整体而言, 这次实验大大提高了我在 C++模板编程和元编程方面的能力。

### 就实验过程中遇到的问题及解决处理方法, 自拟 1—3 道问答题:

#### 1. 模板元编程有什么作用?

答: 可以把一些通常在运行时才能计算的任务提前到编译时, 从而提高程序运行效率或提供一些方便

#### 2. 特化有什么作用?

答: 模板抓住了算法与数据结构上的共性, 但忽略了类型的个性, 设计出的模板对于具体的数据类型而言未必具有最好的效率, 因此定义特殊的实现可以提升效率。

#### 3. 几大排序算法的时间复杂度?

答: 均为  $O(n^2)$

