项目说明文档

数据结构课程设计

——银行业务

作 者 姓 名：安江涛

学　　号：1952560

指 导 教 师：张颖

学院、 专业：软件学院 软件工程

同济大学

Tongji University

目　　录

1　分析.....1
1.1 项目简介.....1
2 设计.....1
2.1 数据结构设计.....1
2.2 类结构设计.....1
2.3 成员与操作设计.....1
3 实现.....2
3.1 重载<运算符.....2
3.1.1 详解.....2
3.1.2 核心代码.....3
3.2 总体系统的实现.....4
3.2.1 总体系统流程图.....4
3.2.2 总体系统核心代码.....5
3.2.3 总体系统截屏示例.....5

1 分析

1.1 项目简介

设某银行有 A，B 两个业务窗口，且处理业务的速度不一样，其中 A 窗口处理速度是 B 窗口的 2 倍----即当 A 窗口每处理完 2 个顾客是，B 窗口处理完 1 个顾客。给定到达银行的顾客序列，请按照业务完成的顺序输出顾客序列。假定不考虑顾客信后到达的时间间隔，并且当不同窗口同时处理完 2 个顾客时，A 窗口的顾客优先输出。

输入说明：输入为一行正整数，其中第一数字 N（N<=1000）为顾客总数，后面跟着 N 位顾客的编号。编号为奇数的顾客需要到 A 窗口办理业务，为偶数的顾客则去 B 窗口。数字间以空格分隔。

输出说明：按照业务处理完成的顺序输出顾客的编号。数字键以空格分隔，但是最后一个编号不能有多余的空格。

2 设计

2.1 数据结构设计

使用 customer 类来存储一个顾客，重载小于运算符后 push 进入 priority_queue，按 priority_queue 的顺序依次输出即可。

2.2 类结构设计

本项目只有一个类，customer 类。本项目用到的 priority 为手写。

2.3 成员与操作设计

customer 类

```
class customer {
public:
    customer() :number(0), time(0), flag(0) {}
    customer(int num, int t, bool f) :
        number(num), time(t), flag(f) {}
    int number;
    int time;
    bool flag;
    bool operator < (const customer& rhs)const {
        if (time < rhs.time)
            return false;
        else if (time == rhs.time && flag)
            return false;
        return true;
    }
    bool operator ==(const customer& rhs)const {
        if (number == rhs.number && time == rhs.time && flag == rhs.flag)
            return true;
        return false;
    }
};
```

3 实现

3.1 重载<运算符

3.1.1 详解

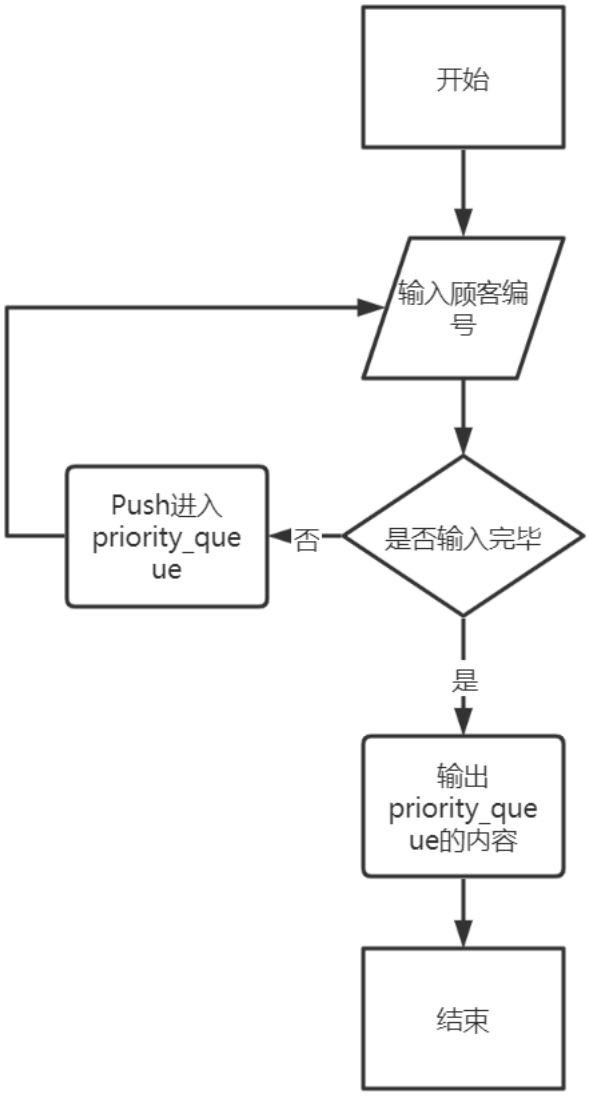
设 x 和 y 是两个顾客，若 x 的输出时间小于 y 的时间，则 x>y;若时间相等，若 x 是 A 业务窗口办理，则 x>y;否则 x<y。因为手写的 priority_queue 是基于大根堆实现的，也就是 priority_queue 里的顺序是从大到小，所以要重载<运算符为>运算符，实现从小打到大排列。

3.1.2 核心代码

```
bool operator < (const customer& rhs)const {
    if (time < rhs.time)
        return false;
    else if (time == rhs.time && flag)
        return false;
    return true;
}
```

3.2 总体系统的实现

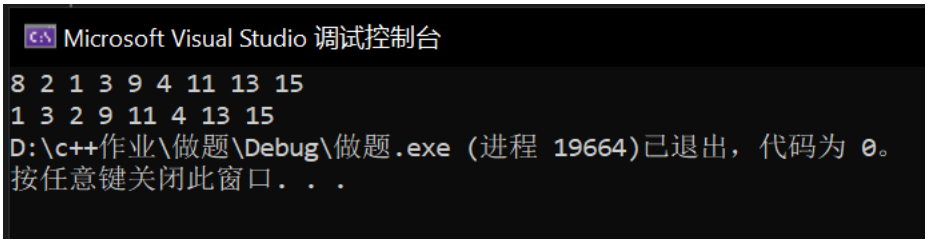
3.2.1 总体系统流程图

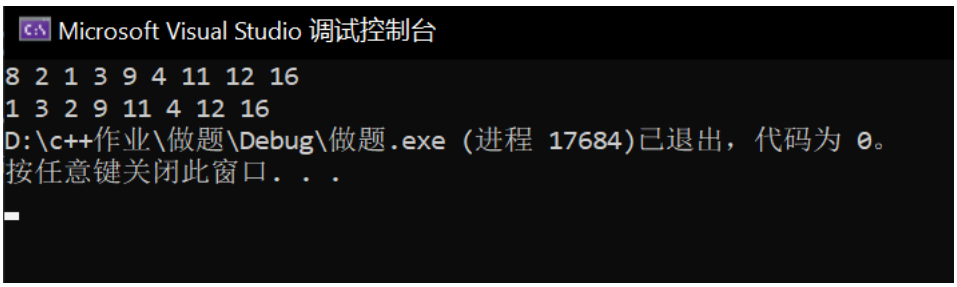


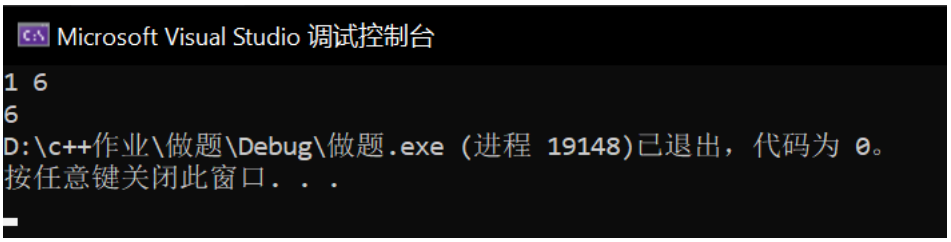
3.2.2 总体系统核心代码

```
int main() {
    int N;
    std::cin >> N;
    int x = 0;
    int time1 = 1, time2 = 2;
    Priority_queue<customer>p;
    while (N--) {
        std::cin >> x;
        if (x & 1) {
            p.push(customer(x, time1, true));
            time1++;
        }
        else {
            p.push(customer(x, time2, false));
            time2 += 2;
        }
    }
    while (!p.empty()) {
        std::cout << p.top().number << ' ';
        p.pop();
    }
}
```

3.2.3 总体系统截屏示例




Microsoft Visual Studio 调试控制台
8 2 1 3 9 4 11 12 16
1 3 2 9 11 4 12 16
D:\c++作业\做题\Debug\做题.exe (进程 17684)已退出，代码为 0。
按任意键关闭此窗口. . .


Microsoft Visual Studio 调试控制台
1 6
6
D:\c++作业\做题\Debug\做题.exe (进程 19148)已退出，代码为 0。
按任意键关闭此窗口. . .