项目说明文档

数据结构课程设计

——修理牧场

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 项目简介

农夫要修理牧场的一段栅栏，他测量了栅栏，发现需要 N 块木头，每块木头长度为整数 L_i 个长度单位，于是他购买了一个很长的，能锯成 N 块的木头，即该木头的长度是 L_i 的总和。

但是农夫自己没有锯子，请人锯木的酬金跟这段木头的长度成正比。为简单起见，不妨就设酬金等于所锯木头的长度。例如，要将长度为 20 的木头锯成长度为 8，7 和 5 的三段，第一次锯木头将木头锯成 12 和 8，花费 20；第二次锯木头将长度为 12 的木头锯成 7 和 5 花费 12，总花费 32 元。如果第一次将木头锯成 15 和 5，则第二次将木头锯成 7 和 8，那么总的花费是 35（大于 32）.

项目功能要求：

（1） 输入格式：输入第一行给出正整数 N（$N<10^4$），表示要将木头锯成 N 块。第二行给出 N 个正整数，表示每块木头的长度。

（2） 输出格式：输出一个整数，即将木头锯成 N 块的最小花费。

2 设计

2.1 数据结构设计

题目要求将一个木头锯成 N 块，那不妨逆向考虑，将 N 块木头合并为一整块木头怎样花费最少，显然这是一道标准的霍夫曼思想的题，但是考虑到大材小用，本项目是用优先队列来处理，其中 Priority_queue 为手写。

2.2 类结构设计

本项目只有一个类，Int 类。主要用于重载运算符。

2.3 成员与操作设计

Int 类

```
struct Int{
    Int()=default;
    Int(int v){val=v;}
    int val;
    bool operator < (const Int&rhs){
        return val>rhs.val;
    }
    bool operator == (const Int&rhs){
        return val==rhs.val;
    }
};
```

3 实现

3.1 重载<运算符和==运算符

3.1.1 详解

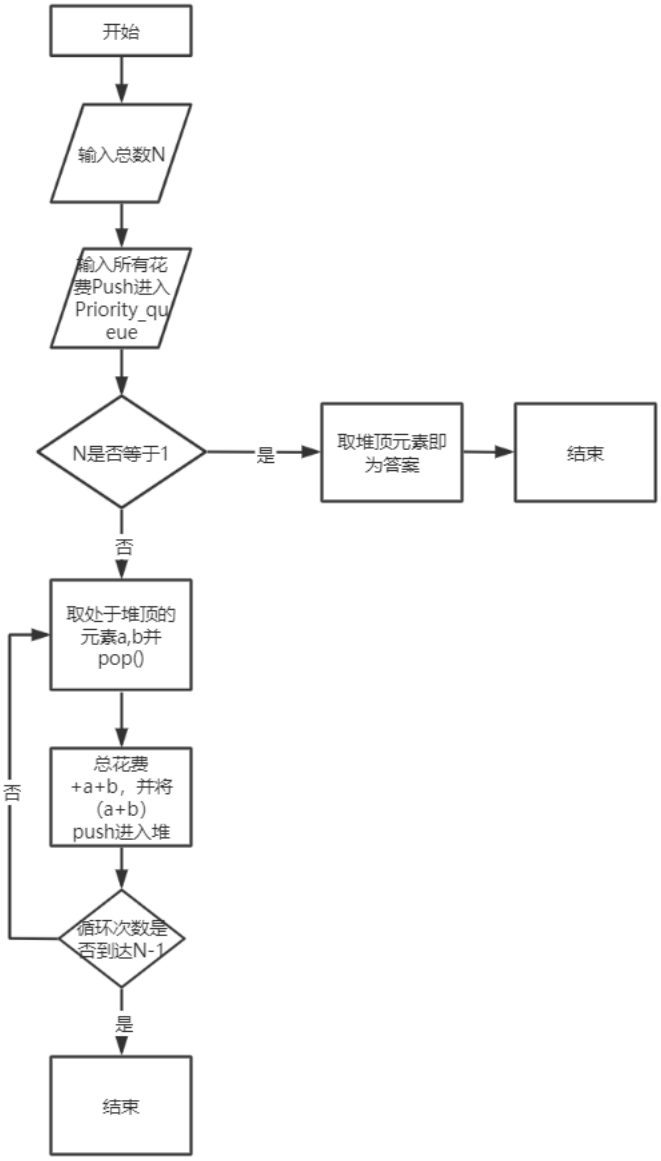
本题要求花费最小，那么显然得用小根堆来实现。考虑到手写的 Priority_queue 为大根堆，且并没有通过模板来传入运算符，那么另外一种方法就是自定义数据结构并重载<运算符以及==运算符。

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;
    while(std::cin.fail()||N<1){
        std::cin.clear();
        std::cin.ignore(INT_MAX,'\n');
        std::cout<<"input error,please re-enput!"<<'\n';
        std::cin>>N;
    }
    Priority_queue<Int> p;
    for (int i = 0, x; i < N; i++) {
        std::cin >> x;
        while(std::cin.fail()){
            std::cin.clear();
            std::cin.ignore(INT_MAX,'\n');
            std::cout<<"input error,please re-enput!"<<'\n';
            std::cin>>x;
        }
        p.push(Int(x));
    }
    int ans = 0;
    for (int i = 0; i < N - 1; i++) {
        int u = p.top().val; p.pop();
        int v = p.top().val; p.pop();
        ans += u + v;
        p.push(u + v);
    }
    if(N==1)
        ans+=p.top().val;
    std::cout << ans << '\n';
}
```

3.2.3 总体系统截屏示例