项目说明文档

数据结构课程设计

——修理牧场

作 者 姓 名： 崔鑫宇

学 号： 1853444

指 导 教 师： 崔鑫宇

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

# 1 分析

## 背景分析

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*Li*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*Li*的总和。但是农夫自己没有锯子，而请人锯木的酬金跟这段木头的长度相等。

## 功能分析

做为计算牧场修理费用的程序，需要的完成的功能是通过收集用户输入的段数以及每段栅栏的长度计算出处理木料的最小费用。

综上所述，经分析，完成计算修理牧场最小费用应该可以通过输入的数字n和接下来的n个数据来计算修理牧场所需的最小费用。

# 2 设计

## 2.1 数据结构设计

如上功能分析所述，完成本功能的程序最适合使用堆这一数据结构。

## 2.2 类结构设计

本程序比较简单只需要一个堆数据结构（Heap）便可达成目的。

## 2.3 成员与操作设计

**堆类（Heap）**

public:

Heap(int size);//堆的构造函数

~Heap();//堆的析构函数

//插入堆中

void insert(int item);

//提取堆中元素

T get();

//调整堆中元素的位置

void OrderDown(int begin, int end);

void OrderUp(int begin);

//得到堆的长度

int length() { return Size; }

//判断队列是否为空

bool isEmpty();

private:

T\* heap;

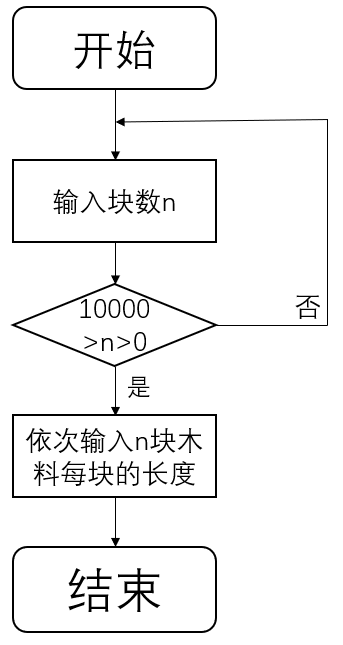
int Size;//堆的大小

## 2.4 系统设计

系统要求用户输入块数n，如果输入不合理会进行提醒。然后由用户自己输入n个正整数，最后自动完成计算工作。

# 3 实现

### 3.1 总体系统流程图



### 3.2 总体系统核心代码

while (heap.length() > 2) {

int \_piece;

int a = heap.get();

int b = heap.get();

\_piece = a + b;

heap.insert(\_piece);

minCost += \_piece;

}

if (num == 1) {//只有一片的情况

int piece\_1 = heap.get();

minCost += piece\_1;

cout << minCost;

}

else if (num >= 2) {//块数大于等于2的情况

int piece\_2 = heap.get();

int piece\_1 = heap.get();

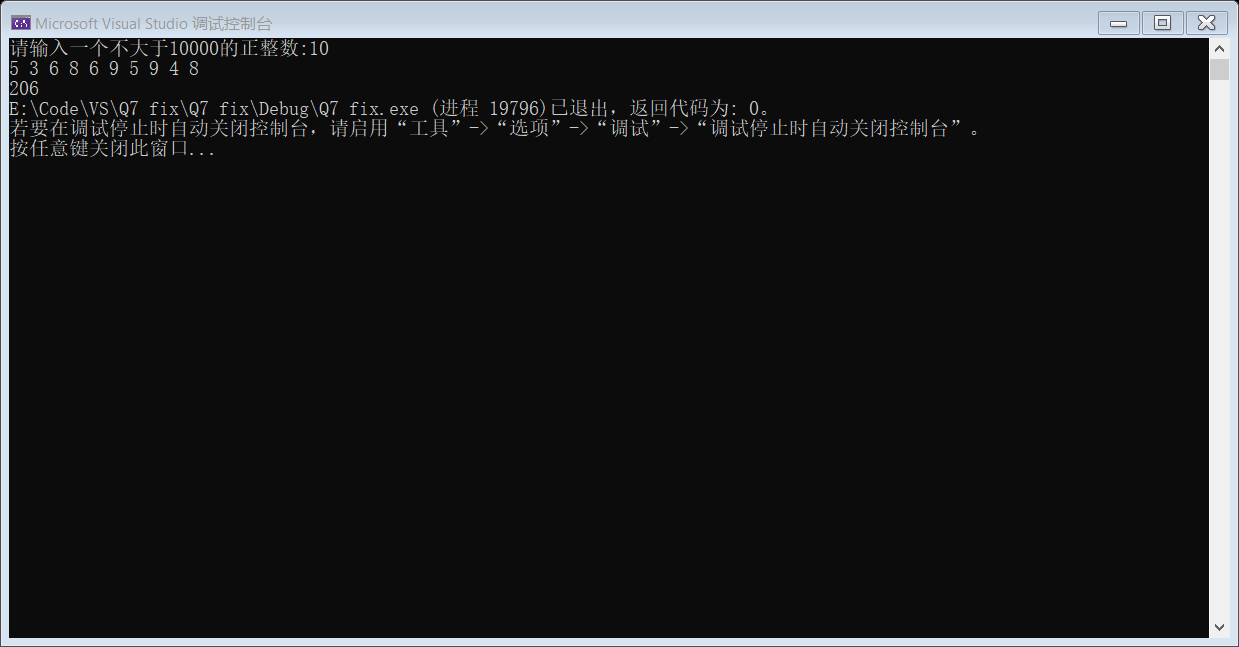
minCost += piece\_1;

minCost += piece\_2;

cout << minCost;

}

### 3.6.3 总体系统截屏示例



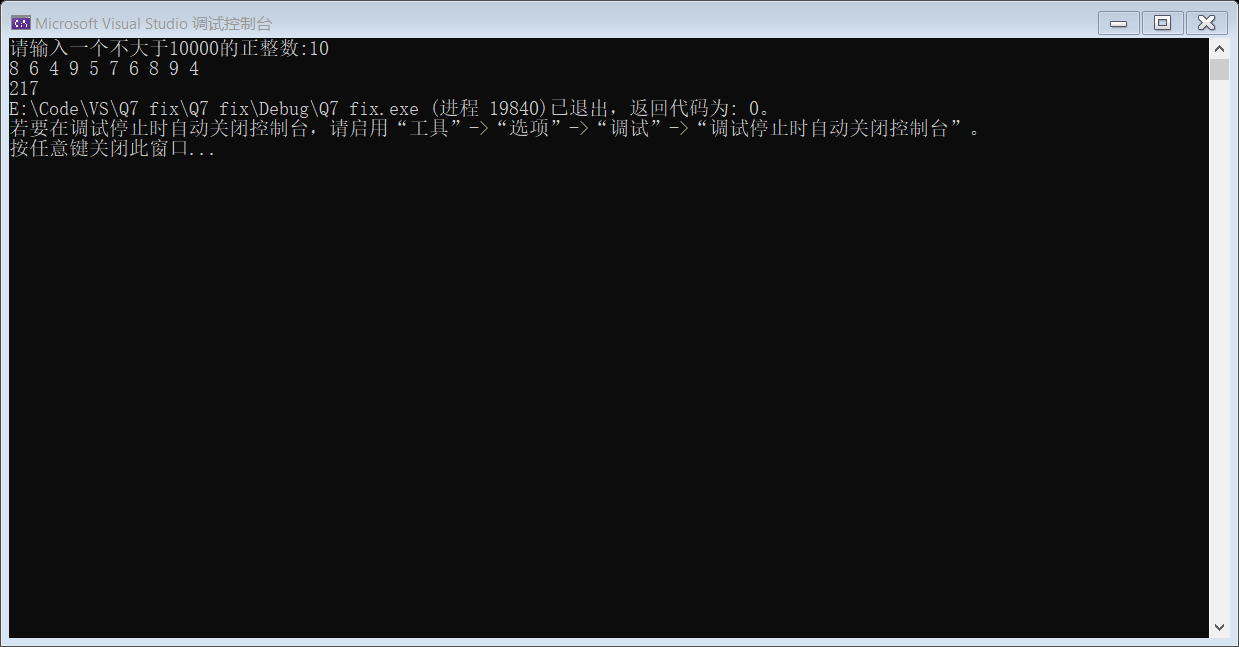
# 4 测试

## 4.1 正常功能测试

**测试用例：**10片，每片长度分别为8、6、4、9、5、7、6、8、9和4

**预期结果：**正常输出

**实验结果：**



## 

## 4.2 出错测试

### 4.2.1 块数为负

**测试用例：**-1

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

****

### 4.3.2 块数大于10000

**测试用例：**10001

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

****

# 