项目说明文档

数据结构课程设计

——修理牧场

作 者 姓 名： 王加炜

学 号： 2150265

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

# 分析

## 1.1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。

## 1.2功能分析

1. 输入格式：输入第一行给出正整数N（N《104），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。

# 二．设计

## 2.1数据结构设计

如上功能所述，由于切割过程是一个由整体到分支的过程，这和树中父节点和子节点的关系类似，所以采用树来进行存储；又因为每个个体具有关键码，所以采用堆的形式存储数据。

## 2.2类结构设计

在堆中，存储了一个一维数组的指针。由于本题注重数据之间的运算而不注重存储每个数据的位置，所以存储形式越简单越好，我决定用数组来存储木头长度。

## 2.3成员与操作设计

### 2.3.1MinHeap类

class MinHeap {

protected:

int num;//实际人数是num+1

int\* heap;

public:

MinHeap()

{

num = -1;

heap = new int[defaultsize];

}

~MinHeap()

{

delete []heap;

}

void siftdown(int start);//从下到上进行调整

void siftup(int start);//从上到下进行调整

bool delmin(int& min);//删除堆顶的最小元素

bool insert(int data\_);//插入一个元素到堆尾

void huffman\_input();//霍夫曼树的输入

void huffman\_output();//霍夫曼树的输出

};

## 2.4类设计

用户输入木条的个数以及每个木条的长度，通过堆的运算，每次操作将堆中最小的和次小的数值取出，然后将两者相加后的结果作为新数加入到堆中。最后循环结束后堆中只剩下一个元素，输出最终的结果。

# **三．实现**

## 3.1信息输入

### 3.1.1文字说明

该函数负责提示用户进行输入总个数和一串数字。

具体实现是首先读入size信息，将size赋值给num，表示该最小堆中目前有size个结点。之后用循环的方式依次读入关键码到堆的相应位置上。同时每读入一个关键码就进行一次从下到上的调整，这样可以保证每个关键码的位置都符合最小堆的特点。

### 3.1.2代码实现

void MinHeap::huffman\_input()

{

int size;

cin >> size;

num += size;

for (int i = 0; i < size; i++)

{

cin >> heap[i];

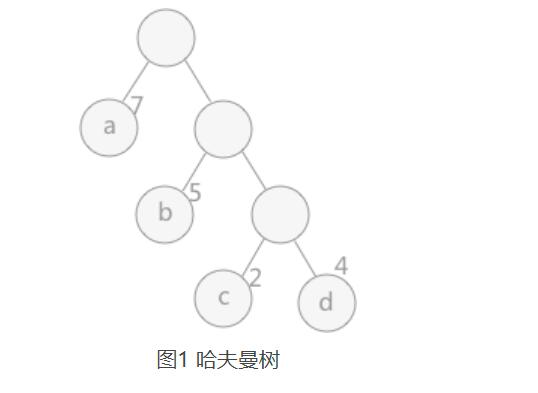
siftup(i);//每输入一个数，就进行一次从下到上的调整

}

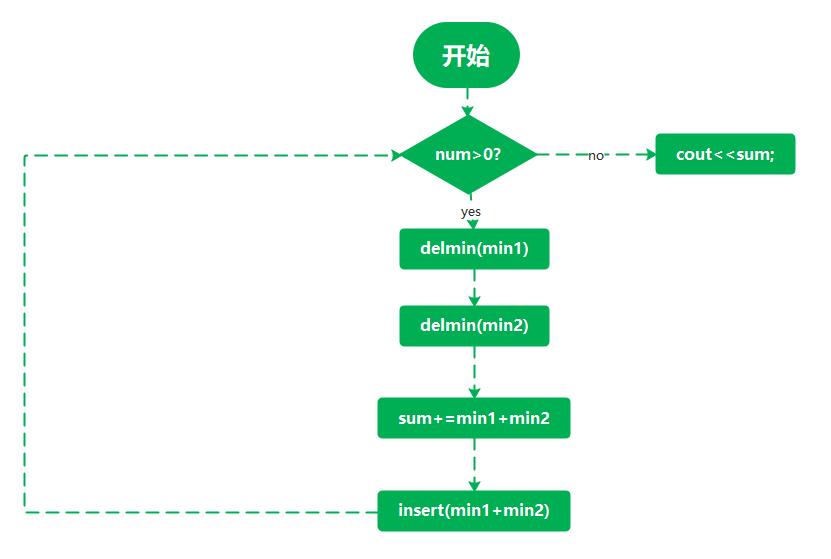
}

## 3.2利用霍夫曼树的构造方式求解

### 3.2.1原理图



### 3.2.2流程图



### 3.2.3文字说明

该函数是本设计的核心函数，负责对于输入数据的处理以及数据的输出。

具体实现是用循环的方式完成的。一次循环相当于模拟一次霍夫曼树的构造过程：删除堆中的最小和次小的元素，然后将两者之和加到sum上，将两者之和作为关键码插入到原堆中。以此进行循环，每循环一次，堆中的元素个数都少一个，直到堆中还剩下一个元素，表示霍夫曼树的构造过程结束。最后将sum输出。

### 3.2.4代码实现

void MinHeap::huffman\_output()

{

int sum=0;

int min1;

int min2;

while (num>0)//num为0是终止条件，表示当前堆中就一个元素

{

delmin(min1);//从堆顶删除最小的元素

delmin(min2);//从堆顶删除次小的元素

sum += min1 + min2;

insert(min1 + min2);//将两者的加和作为新元素插入到堆中

}

cout << sum<<endl;

}

# 四．测试

例子1：

# 1

例子2：

