项目说明文档

数据结构课程设计

——修理牧场

作 者 姓 名： 沈星宇

学 号： 1951576

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

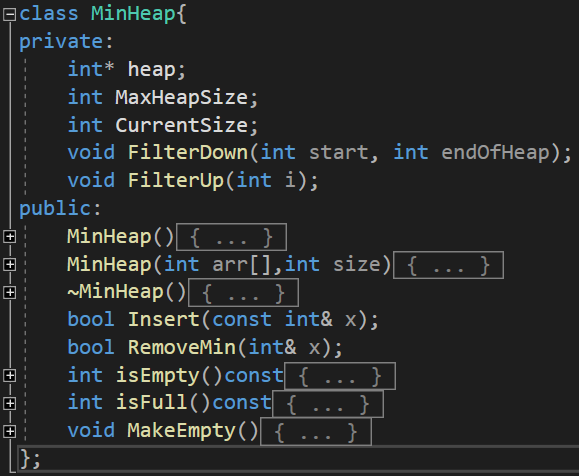
Tongji University

1. **分析**
2. **应用背景**

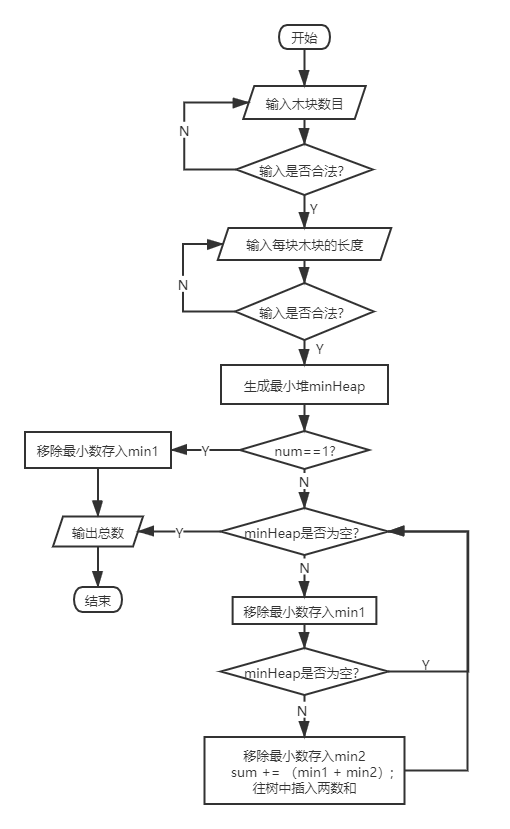
农夫要修理牧场的一段栅栏，他测量了栅栏，发现需要N块木头，每块木头长度为整数Li个长度单位，于是他购买了一个很长的，能锯成N块的木头，即该木头的长度是Li的总和。但是农夫自己没有锯子，请人锯木的酬金跟这段木头的长度成正比。为简单起见，不妨就设酬金等于所锯木头的长度。

例如，要将长度为20的木头锯成长度为8，7和5的三段，第一次将木头锯成12和8，花费20；第二次将长度为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）。

1. **项目功能要求**
2. 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。
3. 输出格式：输出一个整数，即将木头锯成N块的最小花费。
4. **设计**
5. **数据结构设计**



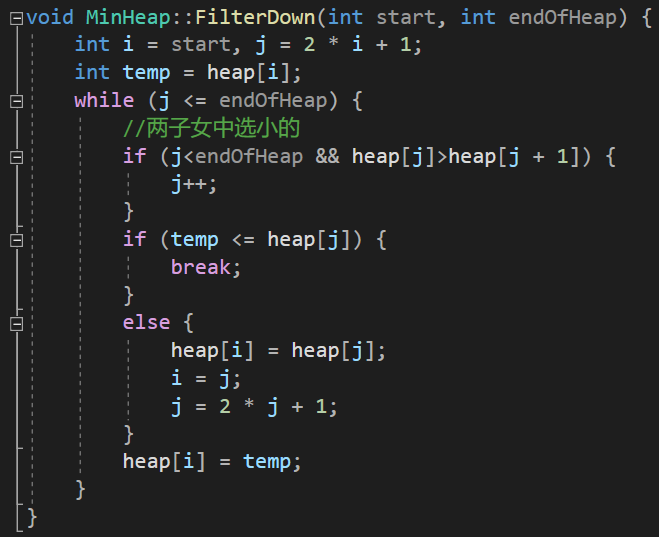
1. MinHeap是最小堆，本项目通过构建哈夫曼树的思想，用最小堆来实现。
2. 函数void FilterDown(int start, int endOfHeap); 是用于自上而下将其调整为最小堆；void FilterUp(int i); 是将插入新节点后的结构从下往上调整为最小堆。
3. 函数bool Insert(const int& x);是往树里插入新的节点，而函数bool RemoveMin(int& x);是将根节点的最小值取出。
4. **程序流程设计**
5. 首先木块的数目，并进行合法性检测
6. 然后输入每块木块的长度，并进行合法性检测
7. 生成最小堆minHeap。
8. 循环调用MinHeap::RemoveMin()函数拿出最小的数字存入min1，若最小堆非空，则再拿出新生成的最小堆的最小数存入min2。将两数相加存入sum，并将两数和重新放回到堆当中。
9. 输出最后的结果sum。

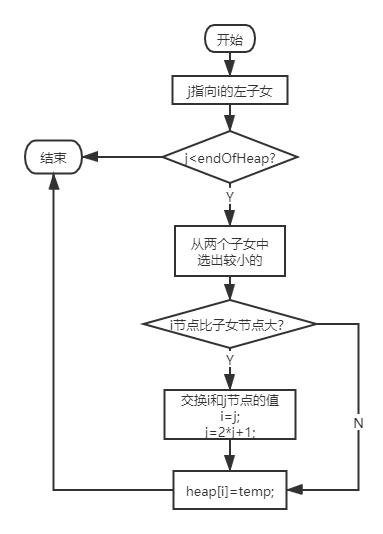


1. **实现**
2. **main()的部分实现**

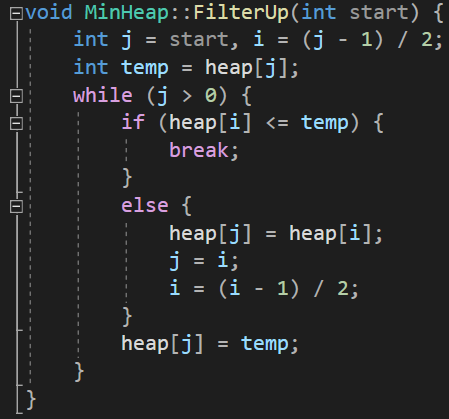


1. 主要利用哈夫曼树的思想，即在构造哈夫曼树的过程中，是将整个序列中最小的两个数字拿出，然后将两数和重新添加到序列当中，那么构造出的哈夫曼树遍历就能得到最小的WPL。此处是用了最小堆来找到整个序列当中最小的数字。
2. 始终要保证再最小堆非空时从里面取数字，循环调用MinHeap::RemoveMin()函数拿出最小的数字存入min1，若最小堆非空，则再拿出新生成的最小堆的最小数存入min2。将两数相加存入sum，并将两数和重新放回到堆当中。
3. **MinHeap::FilterDown()的实现**

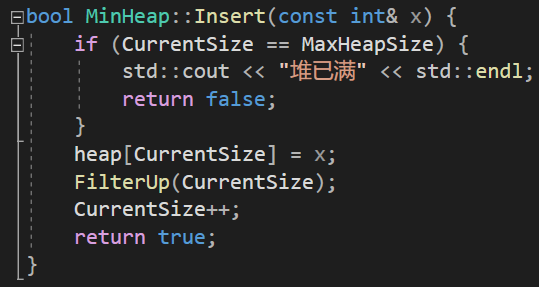




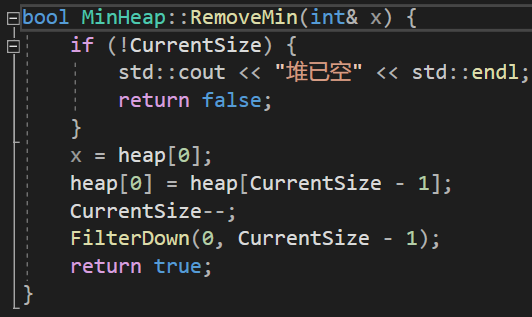
1. 对于一个i来说，j=2\*i+1就是i这个节点的左子女（由堆的性质可得）
2. 当j到达堆的最后一个节点时结束循环
3. 从i的左右子女（如果由右子女的话）当中选出最小的那个，与i节点的值进行比较，如果i节点已经是最小的了，则结束循环；如果i的子女节点比i节点更小，那么交换两个节点的值，同时继续向下搜索（即i指向j，j指向下一个左子女）
4. **MinHeap::FilterUp()的实现**
5. 本函数是针对新加入的节点，从下往上调整它的位置
6. j指向当前节点，i是j的父母节点，循环判断新的节点的值与其父母节点的大小关系，如果新的节点比其父母节点更小，则往上调整其位置；其他的情况则可以退出循环。
7. 基本上就是MinHeap::FilterDown()的逆过程



1. **MinHeap::Insert()的实现**
2. 将插入的值放在数组的最末端，然后调用FilterUp()函数将其调整为最小堆，同时调整最小堆的大小。



1. **MinHeap::RemoveMin()的实现**
2. 移除根节点最小的那个数字，通过引用x来将其值传出
3. 同时对于heap[]数组来说，要做的就是把最前面的数字给放到最后，同时让CurrentSize--,确保该值不会再被访问。
4. 调用FilterDown()函数，调整其重新成为最小堆。

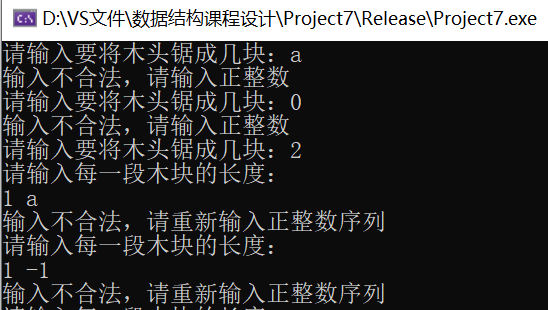


1. **测试**

1、**合法性检测**

（1）输入的木块数目不合法（即非正整数）,会要求重新输入。

（2）输入的木块长度不合法（即非正整数）,会要求重新输入。



**2、一般情况的功能实现**

（1）一般的情况会计算出最小开销

（2）特殊情况要将木头锯成1块（相当于没有锯）那么开销为0

