

Huffman Tree

哈夫曼树

哈夫曼编码

堆

任务要求

完成压缩文件算法的编写

效率

知识点应用

注意事项

合作完成任务

只能使用c/c++语言完成

使用github进行代码同步管理

基本计划

1.第一周

2.第二周

3.第三周

4.第四周

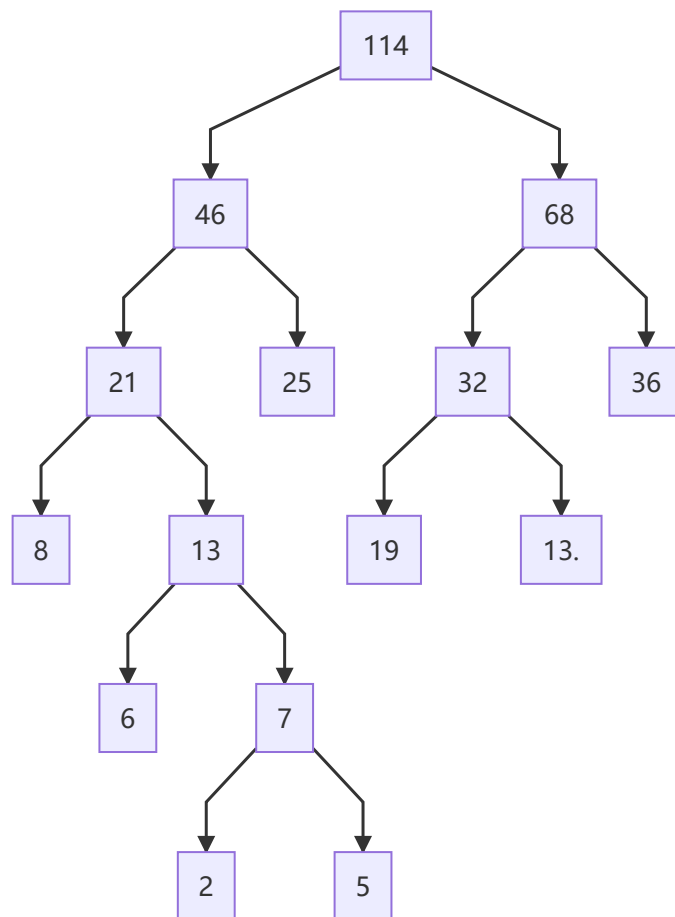
代码审查，答辩

Huffman Tree

哈夫曼树

给定N个权值作为N个叶子节点，构造一棵二叉树，若该树的带权路径长度达到最小，称这样的二叉树为最优二叉树，也称为哈夫曼树(Huffman Tree)。哈夫曼树是带权路径长度最短的树，权值较大的结点离根较近。

如下图就是一颗哈夫曼树。

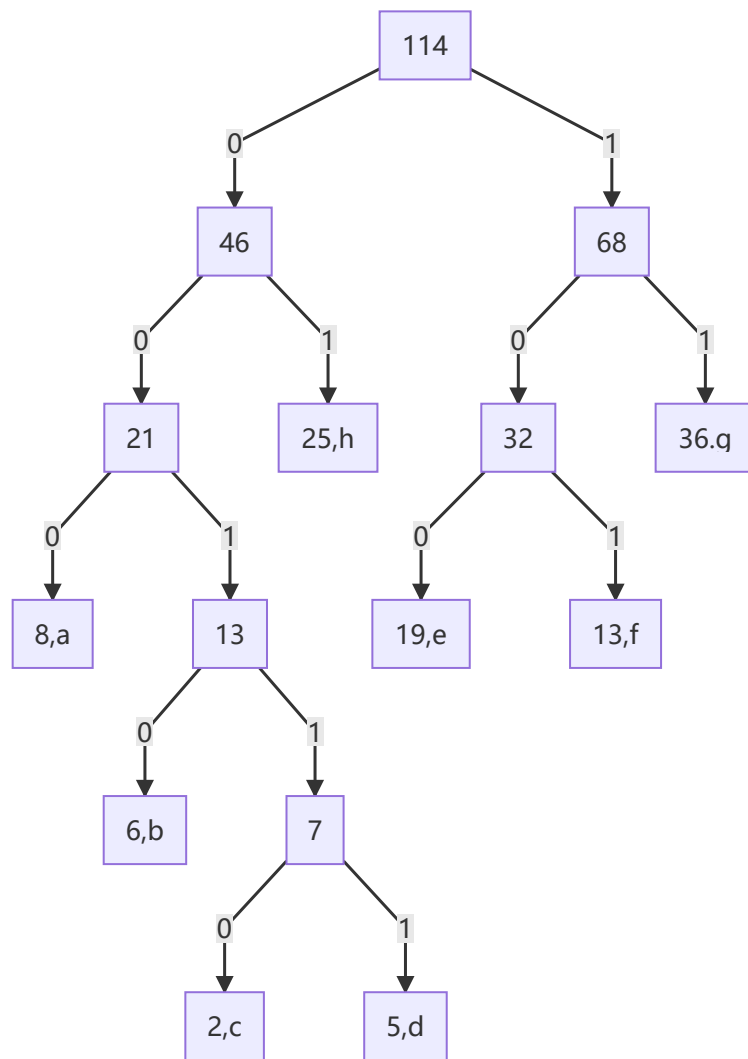


哈夫曼编码

因为哈夫曼树具有越大权重的节点到根部距离越小的特点所以可以用来对数据进行压缩存储。

假设我们有这样一篇文章，其中字母 a, b, c, d, e, f, g, h 出现的次数分别为8, 6, 2, 5, 19, 13, 36, 25次数。如果使用ASCII码进行存储，需要 $114 \times 8 = 912(bits)$ 的存储空间。

建立如下的哈夫曼树。



令树根到各自叶子节点所经过的边作为各自的编码。

得到编码如下

a : 000(8)

b : 0010(6)

c : 00110(2)

d : 00111(5)

e : 100(19)

f : 101(13)

g : 11(36)

h : 01(25)

一共使用的存储空间远小于912bits

堆

堆是一种可以快速找出一堆数据中最大(小)值，删除最大(小)值的数据结构，非常适合用来完成对于哈夫曼树的建立。

任务要求

完成压缩文件算法的编写

要求输入一篇文章，通过上述的方法进行压缩，生成一个压缩文件，然后解压缩重新得到这篇文章，同时保存哈夫曼树的存储结构，方便进行解压缩。

效率

注意编写时算法的效率，需要使用数据结构堆。

知识点应用

这里提示仅供参考。

知识点	描述
1. 数组	数据结构堆中使用数组。
2. 指针	使用指针来增加代码复用性，最好用一段代码通过函数指针建立大根堆和小根堆。
3. 结构体	仔细设计堆的结构体。
4. 文件读写	熟练掌握文件高效读写的方法，在使用w之后对数据进行保存。

注意事项

合作完成任务

- 本次作业由3-5人完成，注意分工配合，每一个函数前用注释加上作者的学号和姓名，格式如下

```
/**
 * @author 2020xxxxxx #?Name
 */
void Quick_sort(int * a,int (*Compare)(const int *a,const int *b)){
    /// something
}
```

- 建议由一位同学完成主要架构，设计函数的接口(参数)，然后由其他同学完成相应的子功能。

只能使用c/c++语言完成

- 在**需要实现的功能**中主要处理流程使用c语言完成。
可以使用数据库等对数据进行保存，数据库的使用这里不提供指导，有需要可以自学。其余工具也一样。
- 操作系统这里不限制为windows，也可以在linux或mac系统上进行代码编写。
- c++禁止使用heap，堆这里需要自己设计实现。

使用github进行代码同步管理

- 各自创建自己的github账号，选出一位主负责人建立项目。
- 其余各位通过fork加入同一个项目进行编写。
- 注意不能两人同时编写同一份文件，由负责人在github上进行代码整合。
- 注意分工，建立良好的项目结构。

基本计划

1.第一周

- 讨论需求，写出要完成的目标到一个文档作为指导。
- 设计项目实施，分工。
- 建立github仓库，学会使用git命令管理github。
- 熟悉github基本操作。
- 建立基本项目框架。

2.第二周

- 各自实现自己的功能，合并代码，调试。
- 不时对需求进行修改，注意最开始设计代码可能会遇到的扩展。

3.第三周

- 实现各自的功能，合并代码，测试。
- 测试，修复bug。

4.第四周

- 深度测试。
- 准备汇报。

以上为预计的基本计划，根据实际需求进行更改。

代码审查，答辩

//最后由相关人员答辩展示代码(可执行文件)功能，每组大约5分钟时间，助教对代码进行审查。