实验三文档

以下两个实验二选一即可！！

Due: 12月31日

一、压缩程序与哈夫曼树

基本描述

你将实现这样一个程序，可以将指定路径的文件(txt,mp3,jpeg,pdf)压缩为以`.huff`为后缀的压缩文件，也可以将以`.huff`为后缀的压缩文件解压成原始文件

基本设计

压缩功能

统计文件中各个char出现的频率，按照书本上的算法构造huffman树，根据huffman树对原文件进行编码，将huffman树和编码后的文件序列写入到`.huff`文件。为了简化基础要求，你可以把文件路径固定在代码中。

Huffman编码可以以“0”或“1”形式存储，对压缩率没有要求。

解压功能

读出对应的huffman树信息，根据huffman树信息解码文件序列，将解码后的信息生成原始文件。

选作内容

(1)使用命令行参数argc与argv读取文件名，你将实现这样一个程序 (+1 分)

```
myhuffman hello.txt
//你将得到hello.txt.huff这个文件
```

(2)使用命令行参数 -z和 -u来区分压缩和解压，需要处理待解压的文件不是压缩文件这种异常情况，你只需打印一条错误信息然后退出程序即可，不需额外的操作。需要处理输入了异常的命令这种情况（例如同时出现-z和-u参数）（+1 分）

```
myhuffman -z hello.txt
//你将得到hello.txt.huff这个文件
myhuffman -u hello.txt.huff
//你将得到hello.txt这个文件
myhuffman -uz hello.txt
//提示错误信息后退出
myhuffman -u hello.txt
//提示错误信息后退出
```

(3)使用命令行参数 -r实现对运行得到的目标文件重命名，例如 （+1 分）

```
myhuffman -zr hello.txt 1.huff
myhuffman -u 1.huff
//解压刚才得到的 1.huff,你应得到hello.txt
```

这一条实际上要求的是同学们思考在压缩原文件时如何将原文件的文件名等信息存储在压缩后的序列中以保证解压这个压缩文件后原文件名不变。

(4)对文件夹进行压缩解压缩。（+2 分）

(5)使用优先队列构建Huffman树，可用Lab 2中定义的优先队列或STL中的优先队列。（+1 分）

(6)将Huffman编码以比特的形式而不是以ASCII的"01"形式存储。(+4 分)

二、导航程序与图上的最短路算法

基本描述

你将实现这样一个程序，设计一个导游程序，能够读取图数据并求两点的最短路径。

基本设计

(1)读取指定文件，用你自定义的图数据结构表示该图。图文件格式为：

该行表示图有3353个点，8870条边。

```
p sp 3353 8870
```

以a开头的行表示第707号点与第1439号点有一条长为40的边。

```
a 707 1439 40
```

以c开头的行是注释

```
c answer 1->4->5->7->2 282
```

你可以按该格式自己设计图来进行调试。

(2)查询任意两个点之间的一条最短的简单路径。要求输出路径和最短距离。

(3)本次实验不对功能菜单形式作过多要求，同学们可以自由发挥。

(4)拿到基础分只需要在江浙沪地图上正确输出结果即可。

选作内容

对导航系统进行优化，使得程序能在 300+MB，2000万左右个点，6000万左右条边的美国数据集上快速运行。

http://users.diag.uniroma1.it/challenge9/data/USA-road-d/USA-road-d.USA.gr.gz

（1）优化最短路径算法的复杂度。平均计算时间在近3年发布的CPU上能控制在10秒以内。（+5 分）

（2）进行数据预处理，减低导航程序的I/O读取时间和初始化图数据结构的时间。

1. 数据集是文本表示的，通过 fscanf 从字符串转换为整数需要大量时间。请将数据集预处理为二进制文件，直接读入内存。（+1 分）
2. 预处理后的二进制文件大小不超过 （2|E|+|N|+2) * sizeof(int)，即562.4MB。（+2 分）
3. 直接读取硬盘数据即可初始化图数据结构，不需要其他操作。初始化时间在固态硬盘上应在1秒以内。（+2 分）
4. 注：满足要求2前必须先完成要求1，满足要求3前必须先完成要求2。

提示：

1. 思考Lab 2 中的优先队列是如何将算法复杂度从 O(n) 降到 O(log n)的。
2. 考虑邻接矩阵如何进行稀疏表示。原始数据集是三元组表示法的，能否想办法改进三元组，既能快速寻找点的邻居，又能减少数据冗余。
3. 为了方便Debug，我们还提供稍小一点的纽约数据集。可以先尝试让其运行时间和初始化时间都控制在50ms以内。