**实验4 数字图像编码实验**

**作业一: 无损编码/压缩算法实验**

问题1: 选择灰度图像，选择合适的排列方式，实现行程编码压缩,计算压缩比；

形成编码就是将图像存储成灰度值+连续相同灰度值的像素个数的数值对的形式，对于明暗变化较多的图像压缩率就会很低，对明暗变化少的图像（比如将图片转化为二值图），压缩率就会很高。在我的实验中，正常灰度图的压缩率是9.626162790697677%，将图片转化为二值图后，压缩率就变为97.93895348837209%。由于是无损压缩，图片还原后和原图是一样的。



问题2: 选择灰度图像，对图像进行8\*8的分块后实现哈夫曼压缩,计算压缩比。

哈夫曼压缩就是用哈夫曼树编码所有灰度值，所以只要实现一个哈夫曼数构建算法就可以了。我借助优先队列对频率进行排序，方便取出和插入元素，每次取出频率最低的两个元素进行合并，再把他们的父节点插入队列，重复操作至队列中只剩根节点，这样树就构建好了。存储时，只需要把灰度值存储成根据哈夫曼树获得的编码即可。最终的压缩率为13.576962209302323 %。由于是无损压缩，图片还原后和原图是一样的。

**作业二: 有损压缩/压缩算法实验**

查阅JPEG编码的有关资料，对图像进行JPEG压缩，算法步骤必须包括如下几个部分：图像分块，离散余弦变换，量化，ac和dc系数的Z字形编排。

问题1: 质量因子分别选为20，60，80，对比显示原图与不同质量因子下解码后的图像；

问题2: 记录图像大小、压缩比、均方根误差，对结果进行分析。

先记录实验结果：

Figure/1.jpg与Figure/1\_20result.bmp的

压缩比为38.1978550666533%

均方根误差为38.725315768826846

Figure/1.jpg与Figure/1\_60result.bmp的

压缩比为39.767302324368037%

均方根误差为78.28939765309043

Figure/1.jpg与Figure/1\_80result.bmp的

压缩比为39.94026253045825%

均方根误差为98.2159363401011

可见，质量因子越大，压缩比越大，同时均方根误差也越大。

JPEG压缩主要包括：图像分割、颜色空间转换、离散余弦变换、数据量化、哈夫曼编码这几个过程。主要产生损失的是数据量化这一步，通过将矩阵中的值除以质量因子后取整的操作减少对浮点数的存储。