1a哈夫曼编码需要有关信息源的先验统计信息，而这样的信息通常很难获得。在多媒体应用中表现得尤其突出，数据在到达之前是未知的，而且符号表的传输本身也是很大的开销。

而自适应的哈夫曼编码则可以解决这个问题。在这种算法中，统计数字是随着数据流的到达而动态地收集和更新的。概率不再是基于先验知识而是基于到目前为止收到的数据。

1b1 bacc

推导过程：

1.接收到0 无操作

2.接收到1 得到01==b，执行下图中第一步的变换

3.接收到0 无操作

4.接收到1 得到01==a，执行下图中第二步的变换

5.接收到0无操作

6.接收到0 无操作，判定输入的是一个新的字符

7.接收到1 无操作

8.接收到0 得到新的字符为c，并执行第三步的变换

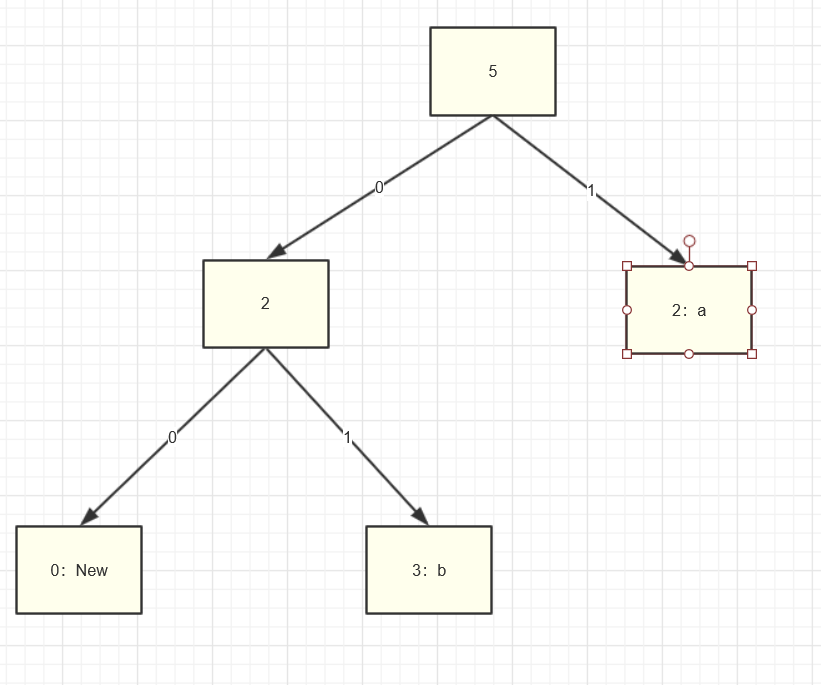
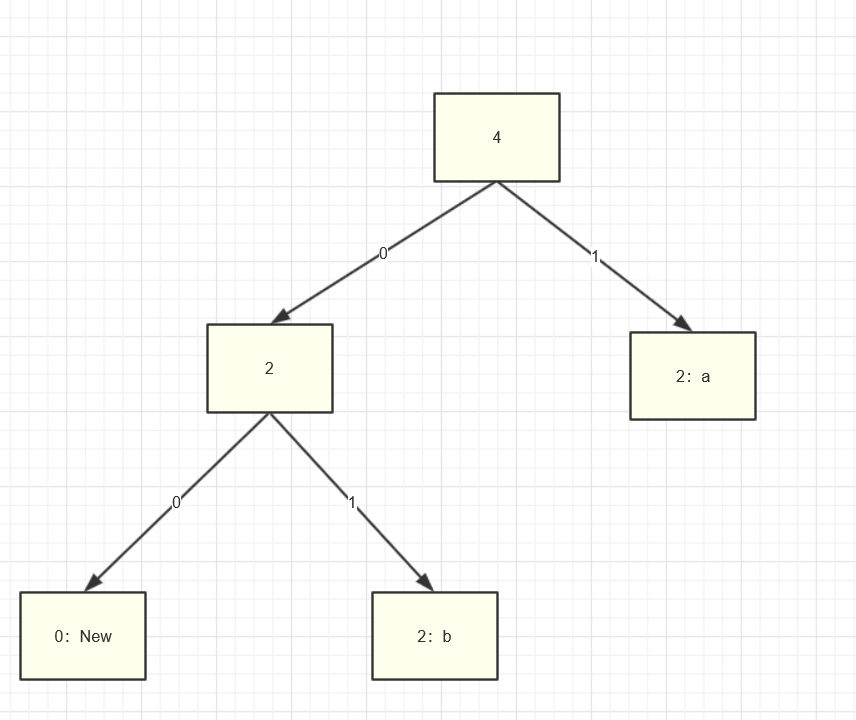
9.接收到1 无操作

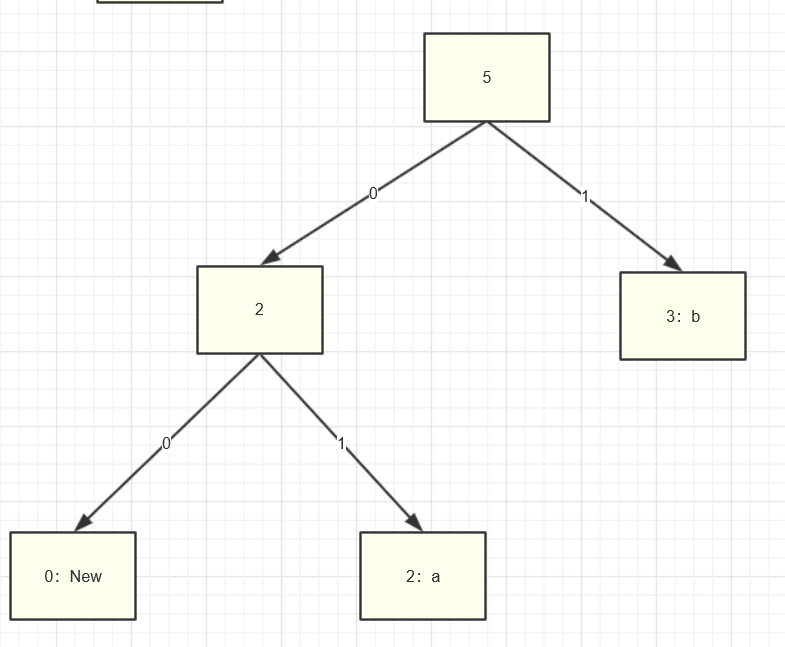
10.接收到0 无操作

11.接收到1 得到101==c，执行第四步的变换

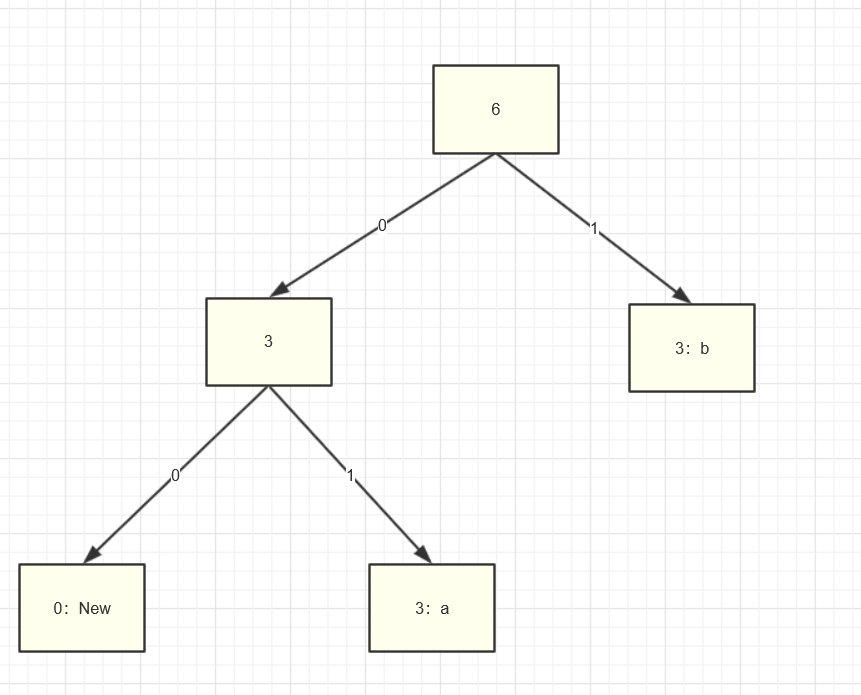
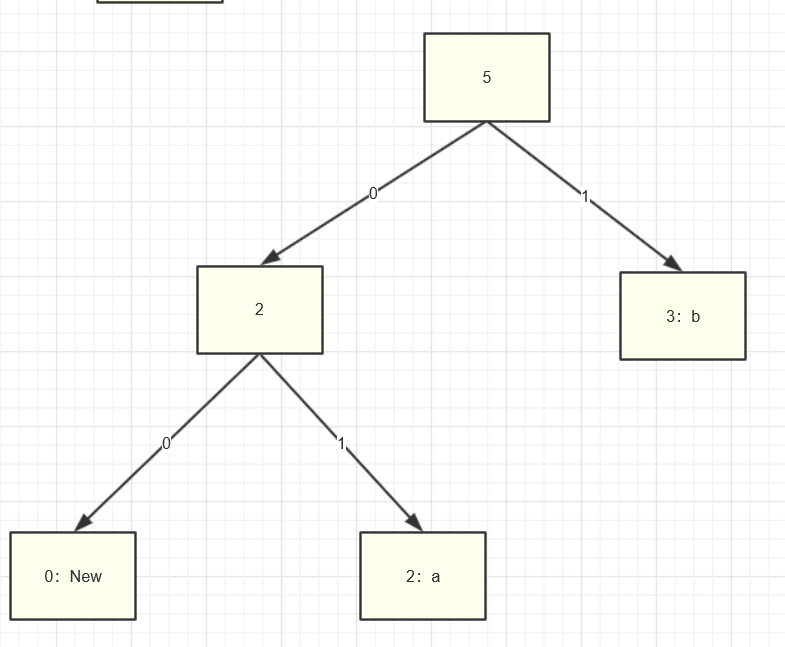
1b2 如图

第一步，接收到b

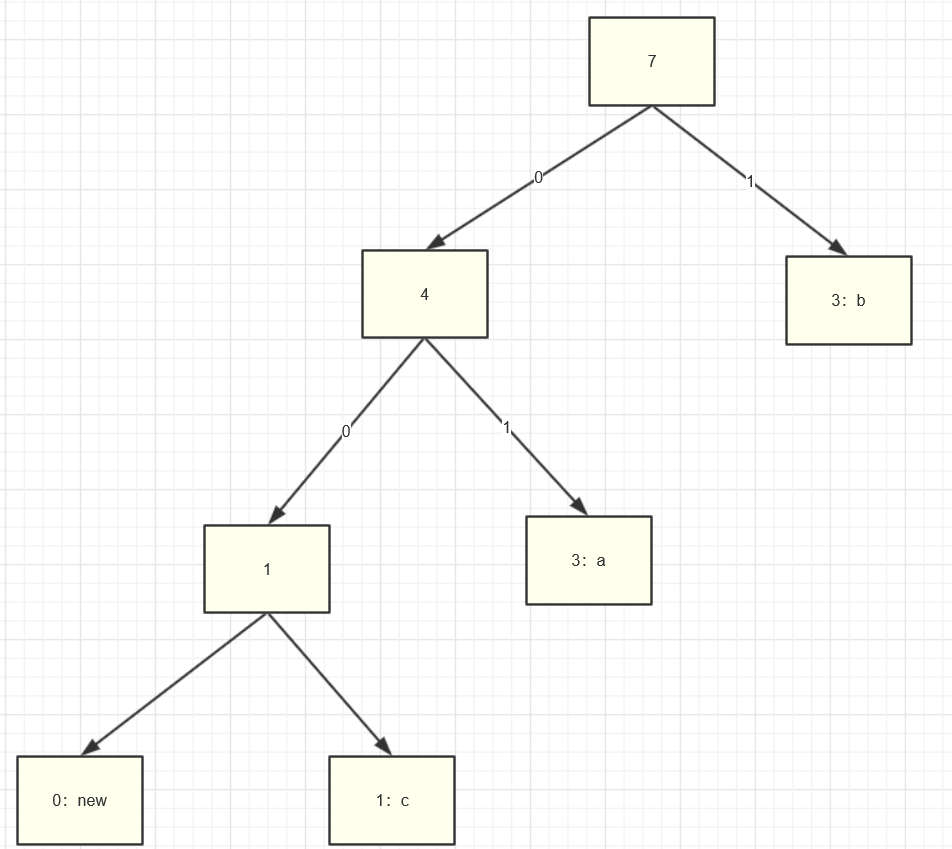
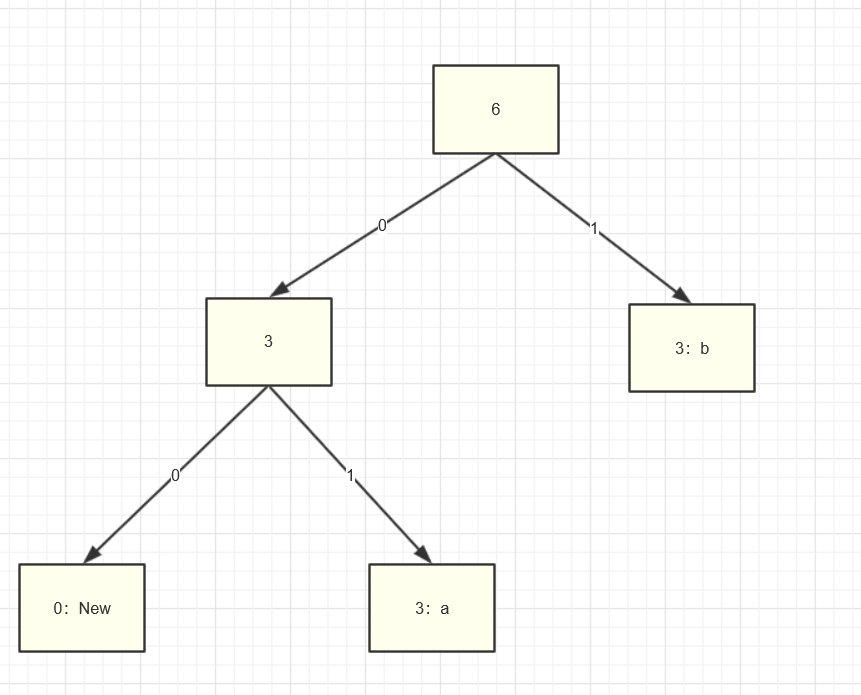


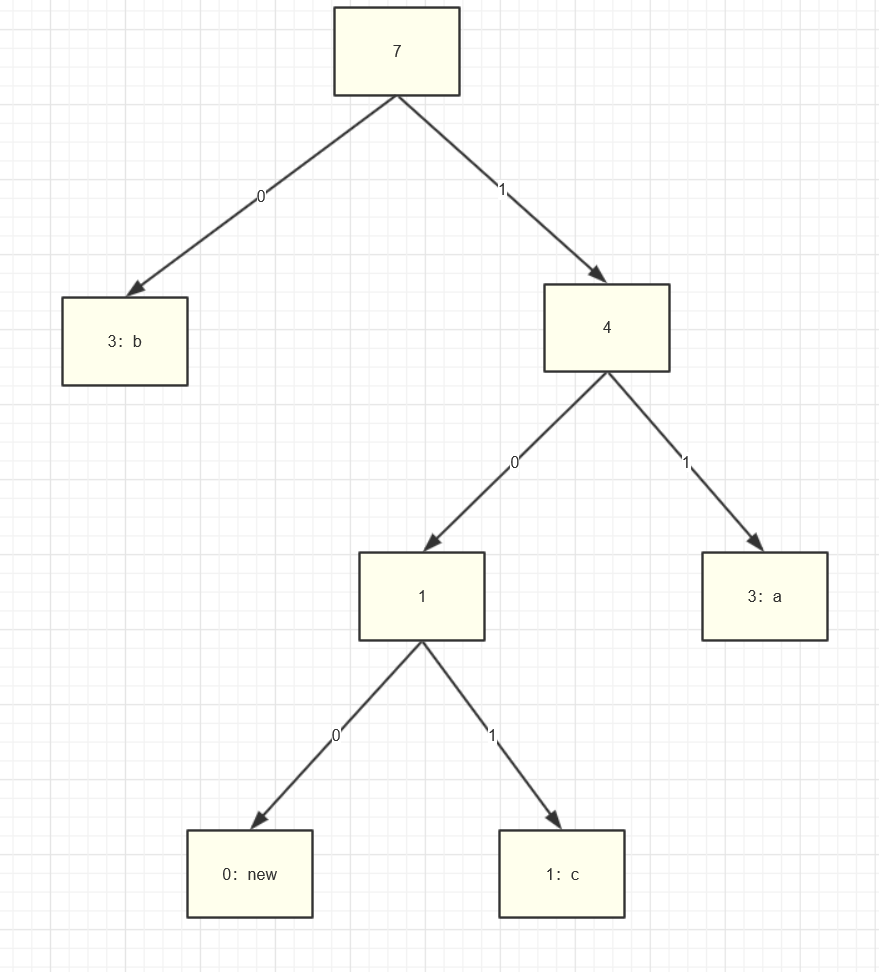


第二步，接收到a

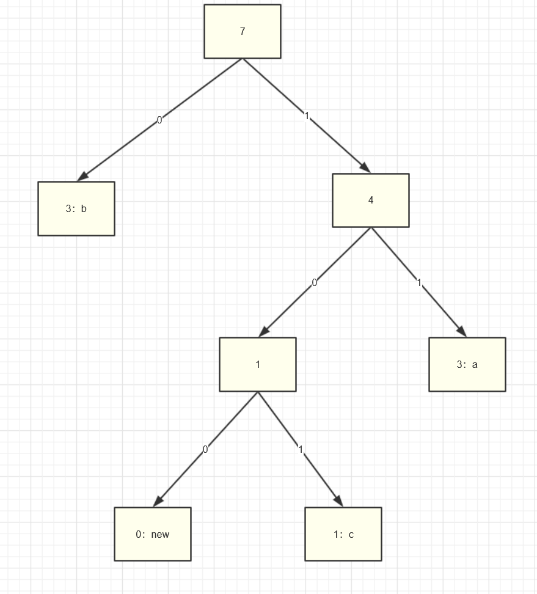


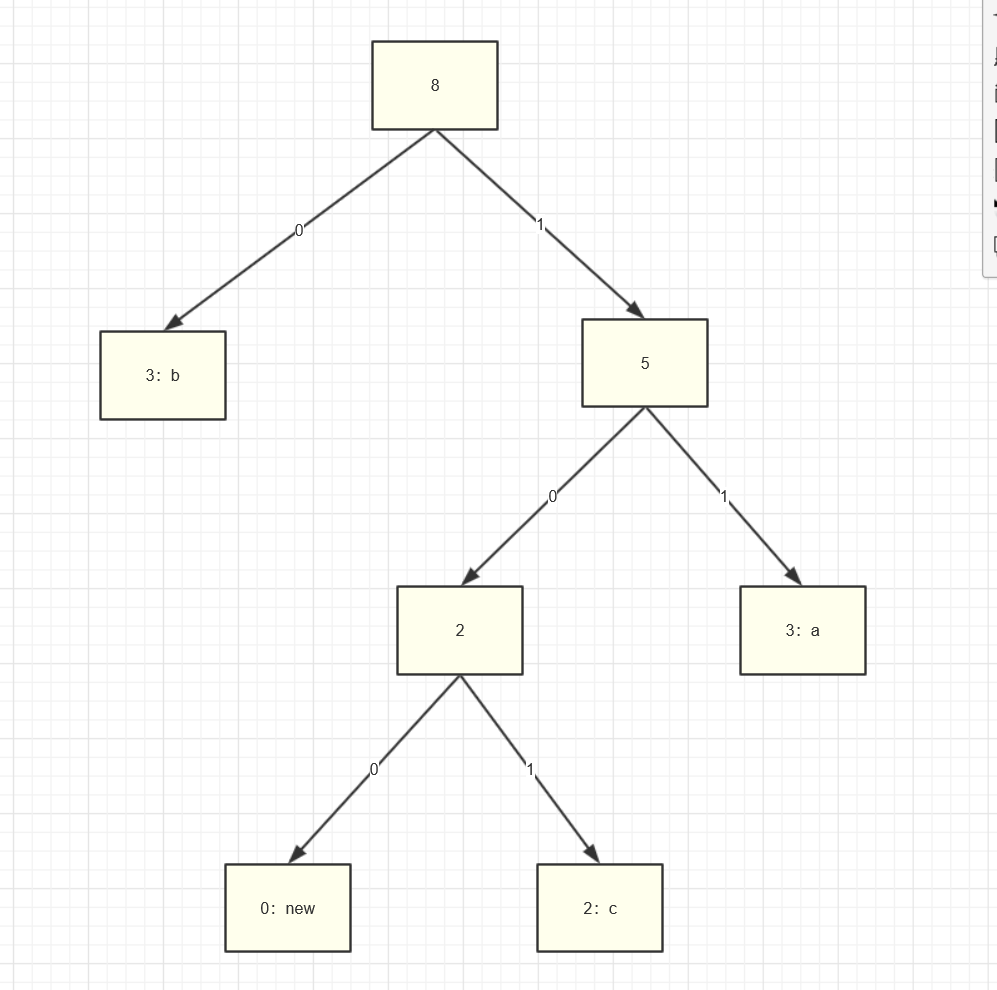
第三步，接收到c





第四步，接收到c





Part2

理论分析：对于色彩丰富的动物图片进行压缩应该选用JPEG，而对卡通图片进行压缩应当选用GIF。由于GIF选用了调色板机制，最多只能存储256种颜色，所以不太擅长处理照片，容易出现大量色块。

GIF加载时可以支持模糊加载（通过隔行载入），使图像更快的出现在屏幕上，而且支持动画，可以把多张图片合为一张成为帧动画。

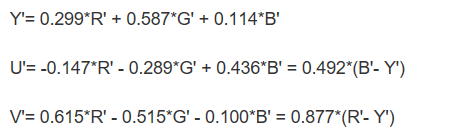
Jpeg可以设置压缩系数来调整压缩的质量，其压缩率更加优秀。JPEG采用了二次采样和DCT变换来丢弃高频信息，并通过熵编码的方法来压缩图像数据，所以Jpeg的压缩能力可以达到非常高。

代码实现部分：代码见JpegSimulator.m 和testhuffman.m

根据题目要求，我实现了

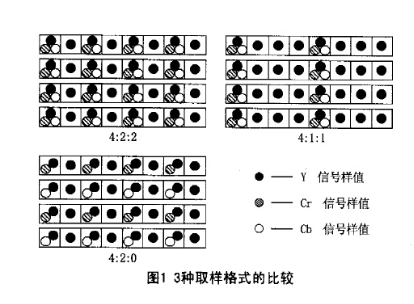
1. 颜色转换

这一个模块把原图的色彩空间从RGB空间转移到了YCrCb上，此处使用了内建的库函数，如果手动实现的话可以使用下面的公式



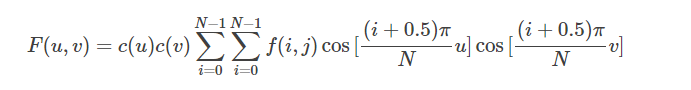
1. 二次采样

在二次采样之前，必须对原图进行格式化，使得行数和列数都能被16整除。格式化可以补0，此处采用的则是全部补边缘列。然后按照4:2:0的比例进行采样。在这一步骤种，信息出现了损失。4:2:0指的是它表示奇数行中，每连续四个点， 采样两个CrCb，在紧邻的第二行中不采样CrCb，而并非是Cb采样为0。



1. DCT变换 见Dct\_Quantize.m

每个图像被划分成了8\*8的块，公



如此，将高频信息和低频信息分离开来。

1. 量化 见Dct\_Quantize.m

将DCT得到的每个频率组成的DCT系数矩阵除以量化矩阵，而后取整。如此，可以将能量集中于低频区域，从而尽可能不损失视觉感觉地减少信息量。量化系数矩阵手动给出，还要再乘上一个质量因子（用来调控图像压缩质量）。接着把所有的量化过后的DCT系数-128。这样做的目的是方便进行游长编码（因为游长编码的压缩通过幅度来进行压缩，这个工序可以使得使用反码可以表示相反数）

1. DPCM编码 见DPCM\_Encoding.m

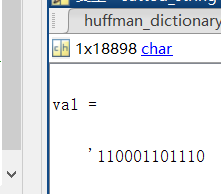
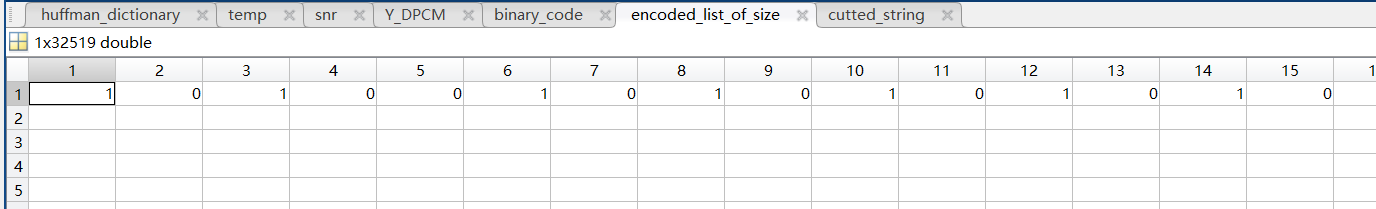
根据课本，这里的DPCM并非有损压缩，而是无损压缩。DPCM的原理是提取每个8\*8的块的第一个数据，然后通过与上一个系数的差来表示本数据。这样一来，数据将由第一个数和接下来的分布在0附近的差值组成。由于集中分布在0附近，可以取得更好的熵编码效果

1. 游长编码 见Run\_Length\_Encoding.m

这一部分是无损压缩中减少数据量最大的一部分。原理是经过量化过的块将体现出能量集中于左上角的特性。我们先进行一次Z扫描（Z扫描通过打表得到），把二维数组改编成一维数组。一般来讲，这个一维数组的波动呈现出前端大后端小的特点，而且越往后0出现的越多。接下来把数组改写成{（本非0数之前的0的个数，本非0数）}这样的集合。注意到第一个数已经存在于DPCM编码中，所以这个集合可以从第二个开始。如果某个块全为0，那么用（0，0）表示它。

1. 熵编码 见testhuffman

这一部分我只完成了一半，对DPCM的编码的结果是一个幅度的位宽的哈夫曼编码，和一个剪切过的幅度值（每个幅度值只保留非0的位宽部分长度）的串。



1. 游长编码解码 见DPCM\_RLE\_Decoding

游长编码的逆过程。通过恢复0的个数把每个串补全64个元素的块（第一个元素是DPCM解码得到的）

1. DPCM编码解码 DPCM\_RLE\_Decoding

DPCM编码的逆过程，比较简单。

1. DCT逆变换

DCT过程的逆过程

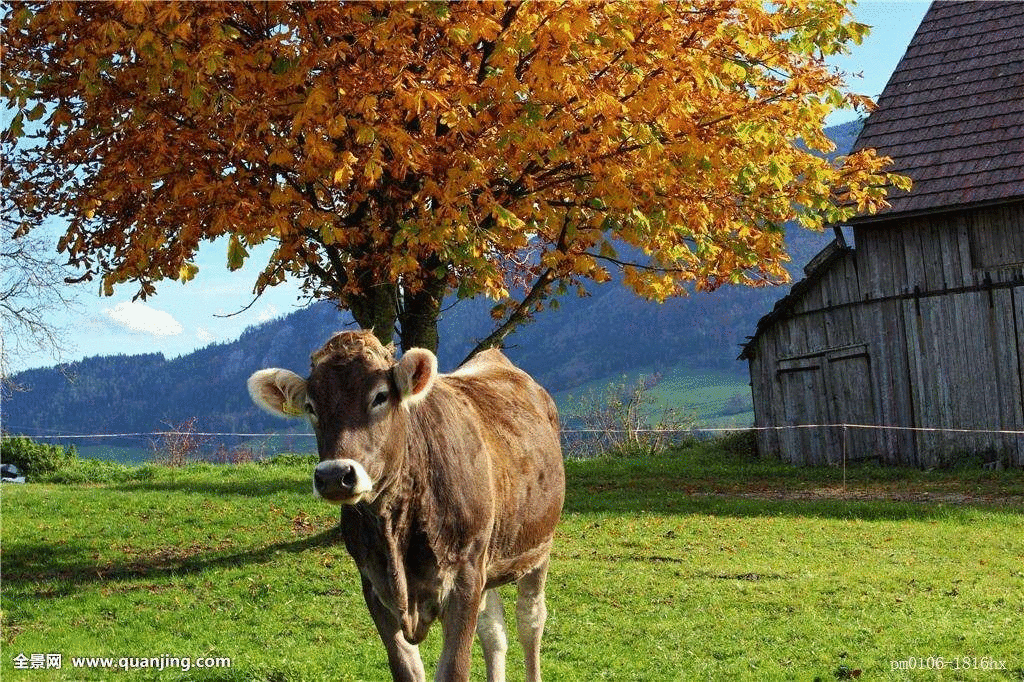
对着公式打，一个字母都不能少

1. 恢复原图

恢复到二次采样之前的数据。

结果对比：可以看到，GIF图片色彩过渡十分生硬，而且有明显的类似喷墨打印机的小点。，在视图中部的草甸上，可以看到明显的色块

和预测不同的是，jpeg在两张图的测试中，无论在图像质量、还是在压缩率上都明显优于gif，并不存在jpeg只在照片情况下优于gif。



gif版本



Jpeg版本



Gif的卡通图



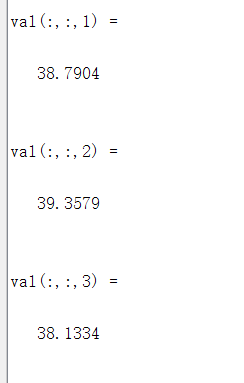
Jpeg的卡通图

压缩率：计算压缩率的时候，不应该以原图作为基准，而应该以原图生成的bmp图像作为原始的数据量，因为原图是jpg格式，本身已经压缩过了。

1. Animal 信噪比通过RGB比较

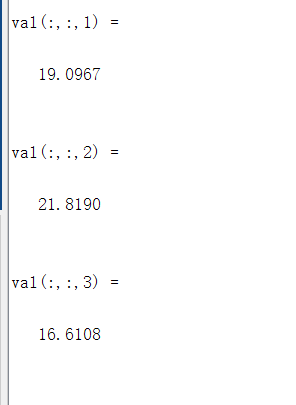
JPEG（Matlab库函数）：

压缩率：230703Byte /2095158Byte=0.1101

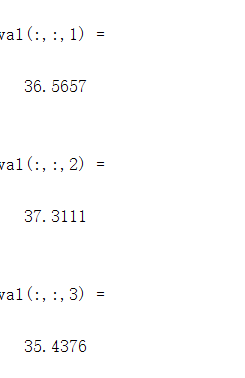
信噪比：

GIF：（来自网站转换器）

压缩率：612336Byte/2095158Byte=0.2923

信噪比：

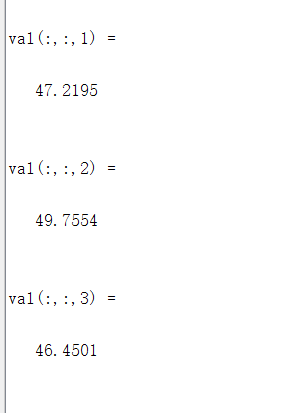
JPEG（个人实现版）：

信噪比：

1. Cartoon 信噪比通过RGB通道比较

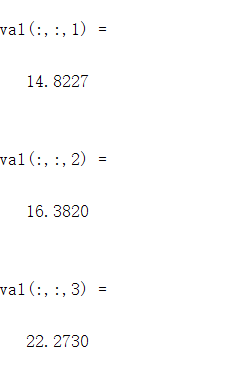
JPEG（Matlab库函数）：

压缩率：119251Byte /2145054Byte=0.0556

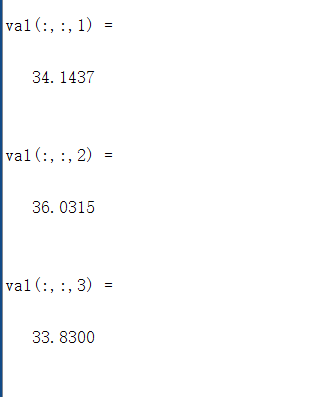
信噪比：

GIF：（来自网站转换器）

压缩率：408312Byte/2145054Byte=0.1904

信噪比：

JPEG（个人实现版）：

信噪比：

代码见附件：运行说明：请运行JpegSimulator，然后才可以运行testhuffman