**17051634张博程**

**注：我的代码都写在一起，运行在最后，我简单地分了些类，有些是测试的，有些是运行的，要运行哪里去掉注释就好。**

作业一：

问题一：

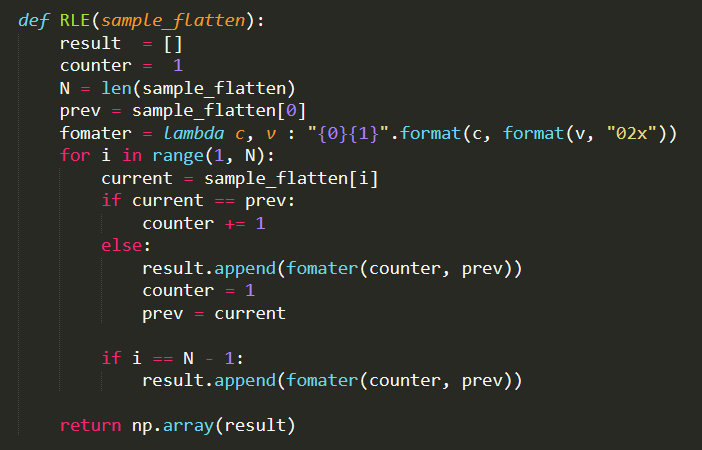
通过检测连续的同一值并简化进行压缩，如：

给出的数据序列为：A-A-A-A-A-B-B-C-D

未压缩前：A-A-A-A-A-B-B-C-D

压缩后：5-A-2-B-1-C-1-D

代码核心：



|  |  |
| --- | --- |
| 图片 | 压缩后大小&压缩比 |
|  | Length After encoding: 1727716  Ratio: 62.4897% |
|  | Length After encoding: 130619  Ratio: 67.6222% |
|  | Length After encoding: 9838588  Ratio: 80.692% |
|  | Length After encoding: 9267673  Ratio: 78.4813% |

总结：

相同色块较多时，压缩比率会相对更大。

问题二：

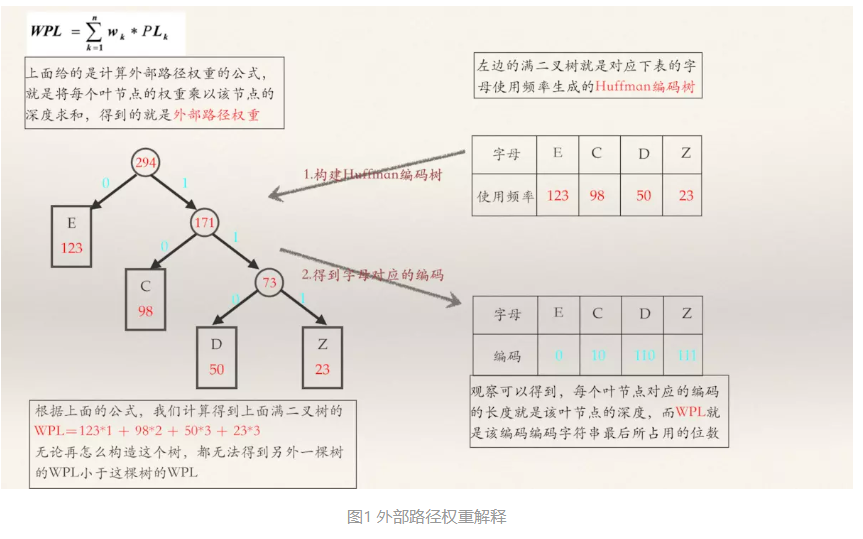
Huffman编码将给字母分配编码。每个字母的编码的长度取决于在被压缩文件中对应字母的出现频率，我们称之为权重。每个字母的Huffman编码是从称为Huffman编码树的满二叉树中得到的。Huffman编码树的每一个叶节点对应于一个字母，叶节点的权重就是它对应的字母出现的频率。使用权重的目的是建立的Huffman编码树有最小外部路径权重。

过程：

 创建n个初始化的Huffman树，每个树只包含单一的叶节点，叶节点纪录对应的字母和该字母出现的频率；

 按照weight从小到大对其进行所有的Huffman树进行排序，取出其中weight最小的两棵树，构造一个新的Huffman树，新的Huffman树的weight等于两棵子树的weight之和，然后再加入到原来的Huffman树数组当中；

 反复上面的2中的操作，直到该数组当中只剩下一棵Huffman树，那么最后剩下来的那棵Huffman树就是我们构造好的Huffman编码树。



|  |  |
| --- | --- |
| 图片 | 大小&比率 |
|  | length before coding:196593  length after coding:196329  ratio:1.0013446816313434 |
|  | length before coding:13776  length after coding:13771  ratio:1.0003630818386464 |
|  | length before coding:2377040  length after coding:2373264  ratio:1.0015910577162928 |
|  | length before coding:2971429  length after coding:2965400  ratio:1.0020331152626964 |

作业二：

步骤一：

图像被分割成大小为8X8的小块，这些小块在整个压缩过程中都是单独被处理的。

步骤二：

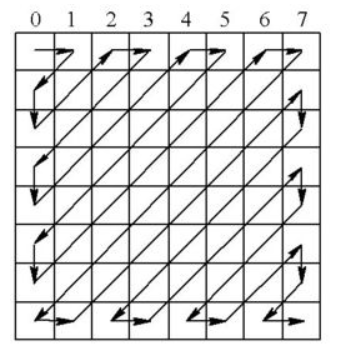
把图案转换成为[YCbCr](http://en.wikipedia.org/wiki/YCbCr)模型

步骤三：

离散余弦变换

步骤四：

数据量化，ZigZag处理，ZigZag编码就是将 8 x 8 的矩阵块按照箭头运动方向重新排列



include文件夹中是编码和解码的各个函数（Fill填充、Sample采样、Block取块、Quantization\_DCT进行DCT并量化、Zigzag排序、Shang\_coding进行DC的DPCM编码和AC的游长编码，并转化为熵编码）。

Quantity取自如下公式：



|  |  |
| --- | --- |
| 图片 | 大小&比率 |
|  | Q=20  before:196593 after:31106 ratio:6.32  Q=60  before:196593 after:50668 ratio:3.88  Q=80  before:196593 after:69714 ratio:2.82 |
|  | Q=20  before: 13776 after:1404 ratio:9.81  Q=60  before: 13776 after:1757 ratio:7.84  Q=80  before: 13776 after:2281 ratio:6.04 |
|  | Q=20  before: 2377040 after:194361 ratio:12.23  Q=60  before: 2377040 after:221738 ratio:10.72  Q=80  before: 2377040 after:264704 ratio:8.98 |
|  | Q=20  before: 2971429 after:264833 ratio:11.22  Q=60  before: 2971429 after:387409 ratio:7.67  Q=80  before: 2971429 after:531562 ratio:5.59 |