

第一题:

8. (a) What are the advantages of Adaptive Huffman Coding compared to the original Huffman Coding algorithm?
- (b) Assume that Adaptive Huffman Coding is used to code an information source S with a vocabulary of four letters (a, b, c, d). Before any transmission, the initial coding is $a = 00$, $b = 01$, $c = 10$, $d = 11$. As in the example illustrated in Fig. 7.8, a special symbol NEW will be sent before any letter if it is to be sent the first time.

Figure 7.18 is the Adaptive Huffman tree after sending letters **aabb**. After that, the additional bitstream received by the decoder for the next few letters is 01010010101.

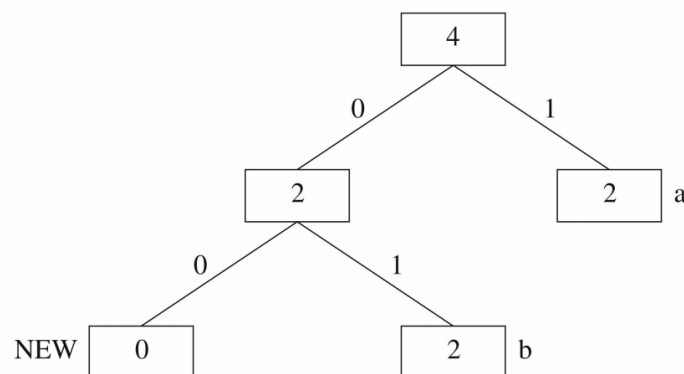


Fig. 7.18 Adaptive Huffman tree

- i. What are the additional letters received?
- ii. Draw the adaptive Huffman trees after each of the additional letters is received.

8.(a)

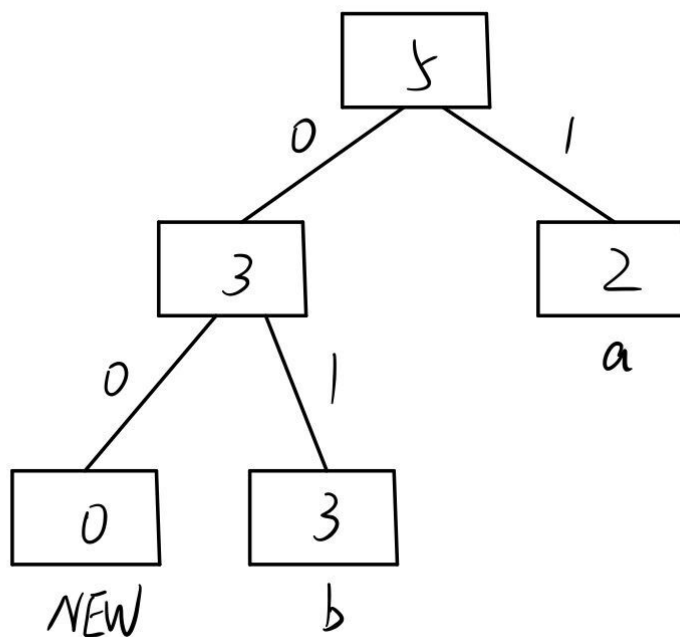
哈夫曼编码需要有有关信息源的先验统计知识，也就是需要提前知道信息源的所有编码信息，是一种全局式的编码算法，但是实际上这样的信息通常是很难获得的，因为数据到达之前都是未知的。同时原始的哈夫曼编码是一种静态的算法，一个哈夫曼表只适用于某一状态的数据，当数据出现一点点变化的时候就不适用了。

自适应哈夫曼压缩算法最厉害的地方在于它的数据是随着数据流的到达而动态收集和更新的，概率不再是基于先验知识而是基于到目前为止实际收到的数据。它允许在符号正在传输时构建代码，允许一次编码并适应数据中变化的条件，即随着数据流的到达，动态地收集和更新每个符号的频率。一遍扫描的好处是可以让程序实时编码；但由于单个数据丢失会损坏整个代码，因此这个算法对于传输错误更加敏感。

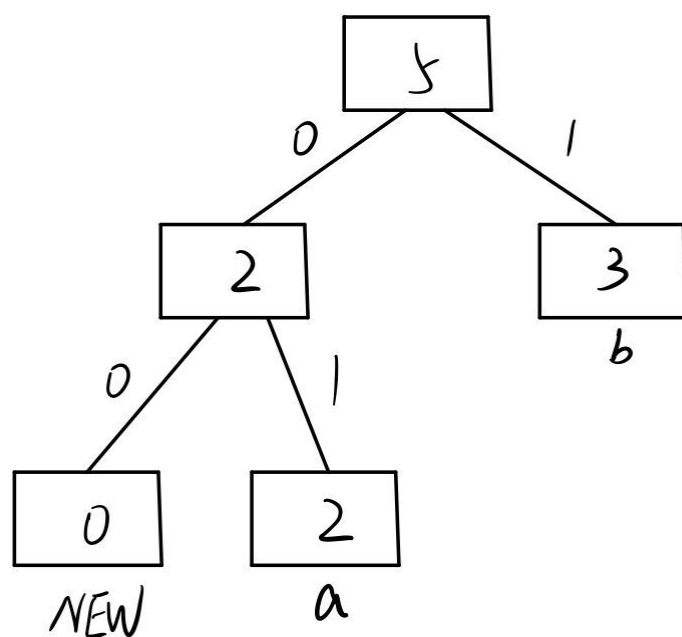
在哈夫曼编码中，有个缺点是除了压缩后的资料外，它还需要传送概率表给解码端，否则解码端无法正确地做解码的工作。如果想要压缩好一点，必须有更多的统计资料给到解压缩端。而自适应哈夫曼编码可以利用已经读过的资料来调整哈夫曼树。

(b)

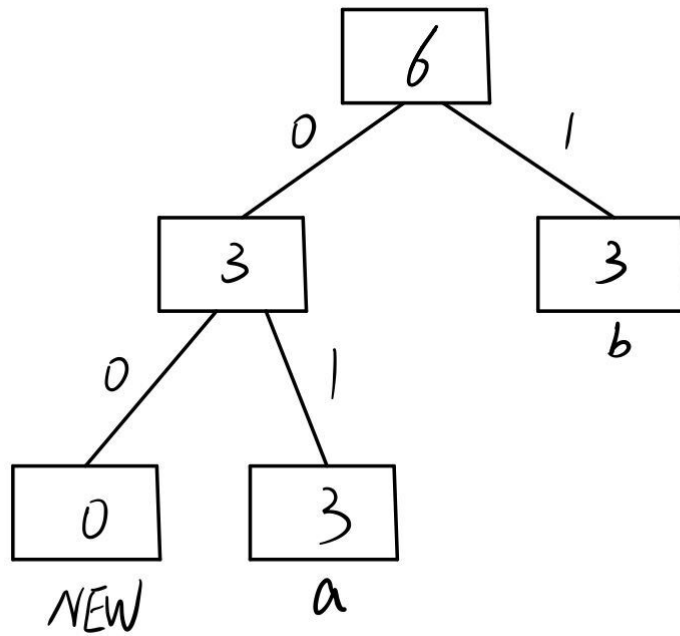
根据所给图可知当前 a , b , NEW 的编码分别为1, 01, 00, 故解码器首先收到的编码为01, 即 b , 哈夫曼树变化为:



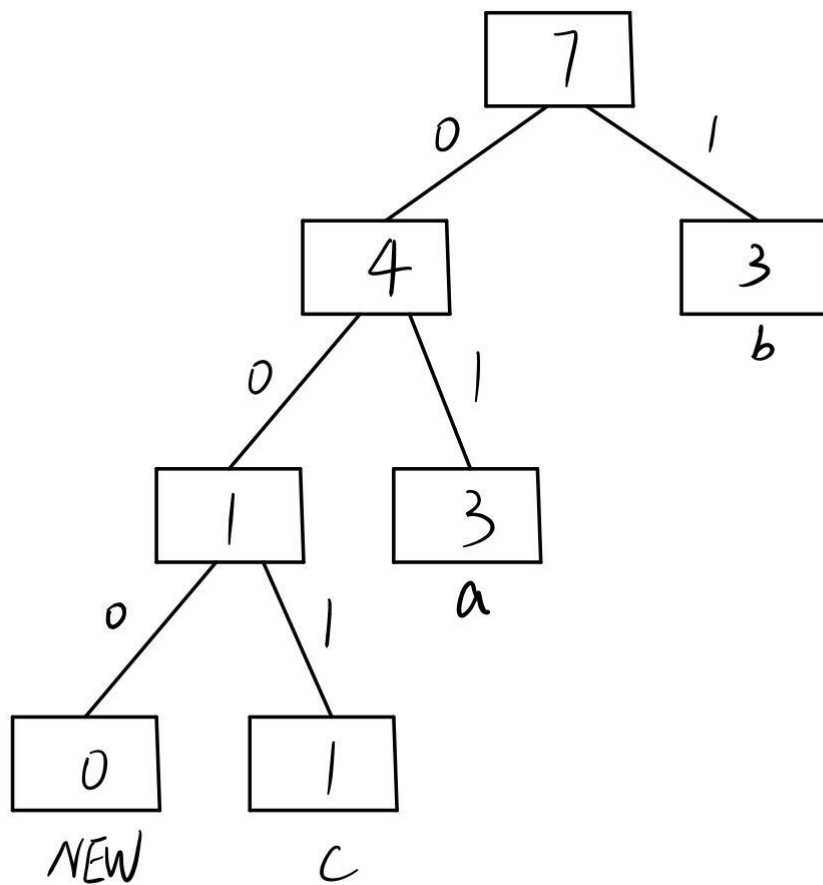
此时需要进行交换:



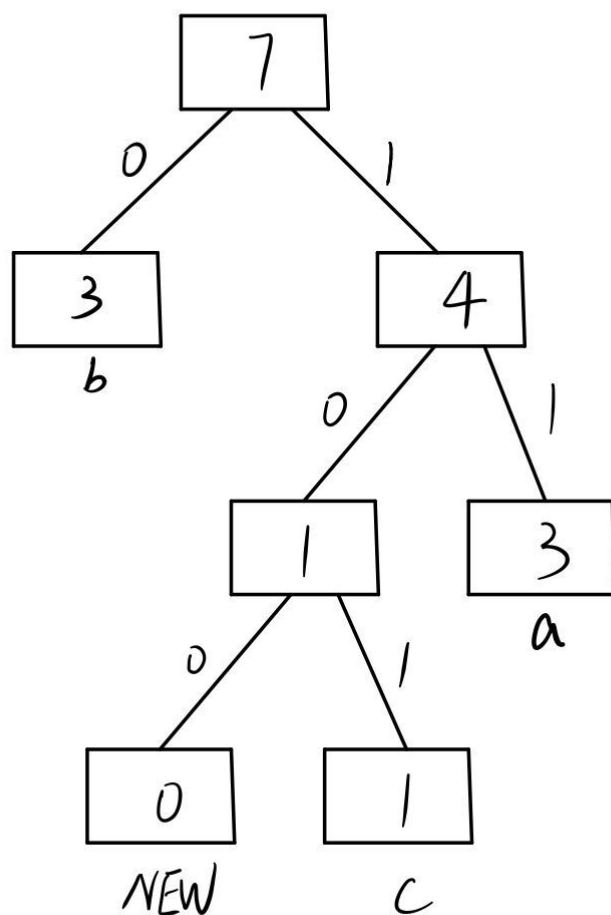
下一个收到的编码为01, 即 a , 故 a 的权重增加1:



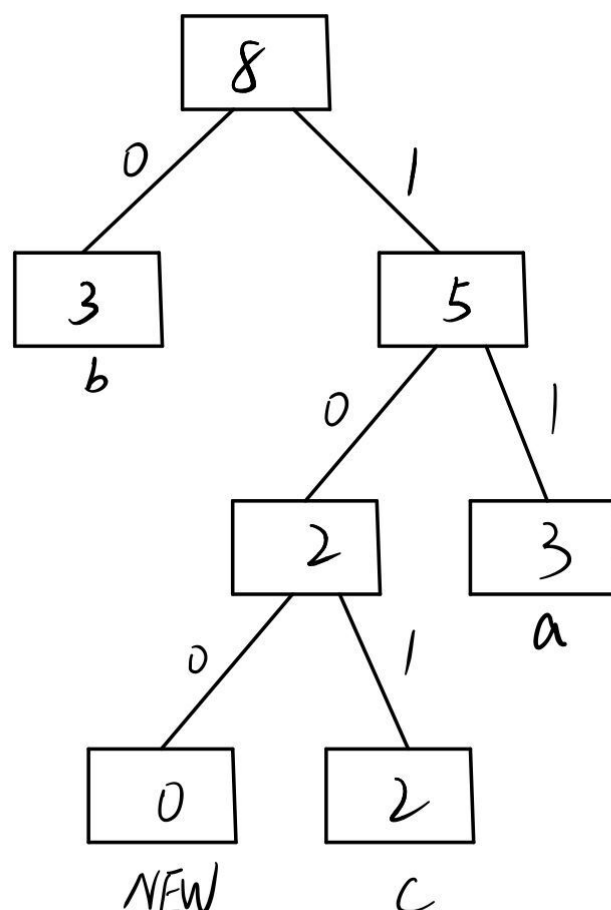
下一个收到的编码为00，即NEW，所以接下来的编码10，即为c，将其添加到哈夫曼树：



此时需要进行交换：



下一个收到的编码为101，即c，哈夫曼树为：



故收到的字母为**bacc**，哈夫曼树已于前面分析过程画出。

第二题：

- You are given a computer cartoon picture and a photograph. If you have a choice of using either JPEG compression or GIF, which compression would you apply for these two images? Justify your answer.

（必须使用本作业提供的图片(动物卡通图片和动物照片)，其他图片无效）

理论原因分析部分（40%），给出你做出选择的原因的解释，包括不同格式图像数据、不同格式图像的编码方法等对比。

程序实现部分（40%）：仅要求实现 JPEG 的压缩算法，需要独立编程实现 JPEG 压缩过程的

重要模块（如颜色转换+色度二次采样，二维 DCT 变换，量化，DPCM 和 游长编码，熵编码等），附上源代码（编程软件不限）；GIF 图像格式可直接用现有软件。

结果对比部分（20%）：JPEG 和 GIF 图像格式的视觉效果和压缩效果比较（包括压缩率的比较和失真度的比较，各占 10%）。

理论分析部分：

JPEG压缩比可以高达100：1，也就是说，JPEG格式的图片在占用很小的空间同时仍然可以保证图片的质量良好。其存储迅速，浏览方便，有着较好的兼容性。但是对于颜色比较少、大块区域颜色相近、有着较大亮度差异的，JPEG压缩无法提供理想的结果。主要原因是JPEG压缩方案可以很好的压缩类似的色调，但是不能够很好地处理亮度的强烈差异或处理纯色区域。

JPEG在DCT与量化两步之后数据就已经集中于左上角，形成的矩阵中含有很多的0，这时候采用游长编码可以降低很多的数据量，同时因为整个图片的DC一般都不会相差很大，所以使用DPCM进行压缩也降低了很多的数据量。最后在前面RLC与DPCM的基础上进行哈夫曼编码使得数据量再次大幅度下降。

JPEG的数据的损失主要是在 DCT变换、量化的步骤中，这是因为在量化的步骤中是进行取整的，所以很多的相似的数据经过量化之后会变成一样的数据，发生归一，在经过逆变换之后这些本不一样的数据就相同了，导致造成了损失。

GIF图片文件的数据是经过压缩处理过的，也就是图片中的RGB值只有256种，相当于在一种256色图片基础上的索引类型的图片。在存储的时候使用一种公共的索引表，并将图片中使用的颜色提取出来，组成一个调色板，而在存储图片点阵的时候只需要存储每一个点在调色板中的索引值。将调色板存储在文件头之中，作为共用的信息。这样的话一个256色的调色盘，24bit的颜色就可以使用8bit来表达了。这样的压缩率为3：1了。另外由于是256色图片，所以颜色总的数目比较少的图片压缩效果会比较好。

GIF压缩是采用无损压缩技术，图像不多于 256 色则既可以减少文件的大小又可以保持质量，普遍用于只需要少量颜色的图像。GIF压缩是将图片转换成 256 色进行传输，256 色中每个颜色都有索引值，然后传输方传输索引值。故GIF适用于颜色数量少、图像细节不明显的图像，因此本题中更适合动物照片的压缩；而JPEG压缩对于颜色多细节多的图像效果会比GIF好得多，更适用于本题中的动物卡通图片。

程序实现部分：

JPEG压缩代码已实现于 `asset/2.py` 文件内，有详细的注释，这里就不再对代码内容进行分析了。









运行环境：Windows10+Python3.6+PILLOW package

运行指令：`python 2.py`

需要注意的是，我采用的方法是先进行编码，再进行解码来获得jpeg图片，与原图进行结果比对。

结果对比部分：

其中 `动物卡通图片.jpg`、`动物照片.jpg` 为助教提供初始图片；首先要将这两张图片用程序转为bmp、gif格式，分别命名为 `cartoon.bmp`、`cartoon.gif` 和 `animal.bmp`、`animal.gif`；`cartoon.jpeg` 为动物卡通图片用编写的JPEG压缩后得到的结果，`animal.jpeg` 为动物照片用编写的JPEG压缩后得到的结果；这些图片已放在 `asset` 文件夹内，可以打开查看。

	<code>animal.bmp</code>	2021/6/3 19:29	BMP 文件	2,047 KB
	<code>animal.gif</code>	2021/6/2 13:46	GIF 文件	584 KB
	<code>animal.jpeg</code>	2021/6/3 19:47	JPEG 文件	223 KB
	<code>cartoon.bmp</code>	2021/6/3 19:28	BMP 文件	2,095 KB
	<code>cartoon.gif</code>	2021/6/2 13:46	GIF 文件	321 KB
	<code>cartoon.jpeg</code>	2021/6/3 20:29	JPEG 文件	117 KB
	<code>动物卡通图片.jpg</code>	2018/11/1 13:35	JPG 文件	114 KB
	<code>动物照片.jpg</code>	2018/11/1 13:37	JPG 文件	173 KB

视觉效果比较：

为了更清晰的对比，截取一小部分进行放大对比，对于这只大象，第一张图为.jpeg，第二张图为.gif，可以看到.jpeg相比于.gif要好很多，.gif里明显可以看到一些噪点（鼻子的粉色区域）还有颜色的偏差（如左上角耳朵的位置）。



压缩效果比较：

根据上面给出文件目录的那张图里的文件大小可以计算压缩率：

动物卡通图片：

JPEG方法： $\frac{117}{2095} = 5.58\%$

gif： $\frac{321}{2095} = 15.32\%$

动物照片：

JPEG方法： $\frac{223}{2047} = 10.89\%$

gif： $\frac{584}{2047} = 28.53\%$

对于两张图片，显然，JPEG压缩率更低，压缩的更好。

这里用 MSE 来衡量失真度，用最一开始助教所给图片和压缩后的.jpg，.gif图片来进行对比，代码如下：

```
import numpy as np
from PIL import Image

def compute_mse(X, Y):
    x = np.float32(X)
    y = np.float32(Y)
    mse = np.mean((X - Y) ** 2, dtype=np.float64)
    return mse

if __name__ == '__main__':
```

```
# 将两张图片都转成灰度图进行比较
img1 = Image.open("cartoon.gif")
# img1 = Image.open("cartoon.jpeg")
# img1 = Image.open("animal.gif")
# img1 = Image.open("animal.jpeg")
img1 = img1.convert('L')
img2 = Image.open("动物卡通图片.jpg")
# img2 = Image.open("动物照片.jpg")
img2 = img2.convert('L')
print(compute_mse(img1, img2))
```

可以得到结果：

动物卡通图片：

JPEG方法：1.85

gif：19.54

动物照片：

JPEG方法：22.00

gif：19.65

对于动物卡通图片，显然，JPEG的失真度更低，效果比gif要好很多。但是对于动物照片，则是gif的更小，不过JPEG的失真度与gif的很接近。

根据这几组数据我们可以看到，对于类似动物卡通图片这种细节较多、颜色比较多样的，JPEG更适合于gif；对于类似动物照片这种颜色比较少的，二者的效果接近，有时gif可能会更优。这也印证了我们之前的理论分析部分。