ode + temp2

code = code + last[i]

while code.\_\_len\_\_() % 8 != 0: # 因为一个字节为8bit，最小单位为字节

code = code + '0'

code\_int = int(code, 2)

code\_hex = hex(code\_int)

code\_hex = code\_hex.lstrip('0x')

code\_byte = bytes.fromhex(code\_hex)

return code\_byte

def lz\_decode\_dic\_make(binary\_str):

num = [] # 组号表

last = [] # 码字表

i = 0 # 用来判断读取完成没有

j = 1 # 段号

k = 0 # 用来判断读了几位了

code\_lenth = 1

str1 = ''

str2 = ''

while i < binary\_str.\_\_len\_\_():

k = k + 1

str1 = str1 + binary\_str[i]

temp1 = math.log(j, 2)

temp1 = math.ceil(temp1)

if j == 1:

temp1 = 1

if k < temp1 + code\_lenth:

str2 = str2 + binary\_str[i]

i = i + 1

continue

else:

num1 = int(str2, 2)

# print(num1)

last.append(binary\_str[i])

num.append(num1)

k = 0

j = j + 1

i = i + 1

str1 = ''

str2 = ''

dic = []

dic.append('')

i = 0

code = ''

while i < num.\_\_len\_\_():

temp3 = dic[num[i]] + last[i]

dic.append(temp3)

i = i + 1

return dic

def lz\_encode(source\_file,res\_text):

binary\_str, text\_len = file\_to\_bit(source\_file) # 文件转换为二进制字符串

bytes\_len = inversion(text\_len)

bytes\_len = int(bytes\_len, 2)

print("source file bytes:", bytes\_len)#输出源文件大小

num,dic,last=lz\_encode\_dic\_make(binary\_str) #建立lz的字典

code\_byte=lz\_encode\_code\_make(num,dic,last) #建立lz编码

img2 = open(res\_text, 'wb')

img2.write(code\_byte)

img2.close()

print("encoded file bytes:", code\_byte.\_\_len\_\_())#输出压缩文件大小

def lz\_decode(source\_file,res\_file):

binary\_str, text\_len = file\_to\_bit(source\_file) # 文件转换为二进制字符串

bytes\_len = inversion(text\_len)

bytes\_len = int(bytes\_len, 2)

print("encode file bytes:", bytes\_len)#输出压缩文件大小

dic=lz\_decode\_dic\_make(binary\_str)#从二进制字符串中提取出lz字典信息

#从建立好的字典恢复源码

code = ''

i = 1

while i < dic.\_\_len\_\_():

temp4 = dic[i]

code = code + temp4

i = i + 1

code\_int = int(code, 2)

将文件作为信源，对文件进行压缩是通过信源编码实现的。由于信源符号之间存在分布不均匀的相关性，使得信源存在冗余度，信源编码的主要任务就是减少冗余，提高编码效率。对于计算机中存储的文件，其大多是按二进制形式存储的，即，它们已经对原文件信息进行了二进制编码，并把信息按二进制编码存储。但由于这些文件的自身特性，如考虑通用性或易操作性而采用定长编码或整字节编码，导致其并未完全按照信源符号概率分布进行编码，编码效率较低，进而使存储时有较大冗余。

为了能无失真还原文件，我们使用了无失真的Huffman编码进行文件压缩。以下将简要介绍Huffman编解码原理。

1.1 Huffman编码

Huffman编码是一种变长分组编码，完全依照各字符出现的概率来构造码字，是一种统计匹配编码。其基本原理是基于二叉树的思想，所有可能的输入符号在Huffman树上对应一个节点，节点的位置就是该符号的Huffman编码。为了构造唯一可译码，这些节点都是Huffman树上的叶子结点。

为了构造Huffman树，首先须知道所有信源符号的概率。具体编码方法如下：

（1） 将信源消息符号按其出现的概率大小依次排列。

（2） 取两个概率最小的符号分别配以0和1两个码元，并将这两个概率相加作为一个新符号的概率，与未分配二进制符号的字母一起重新排队。

（3） 对重排后的两个最小符号重复步骤（2）的过程。

（4） 不断继续上述过程，直到最后两个符号配以0和1为止。

（5） 从树根开始，向下返回得到各个信源符号所对应的码元序列，即相应的码字。

Portal中的原有数据将保存到2006年11月30日，请通过ftp://oldportal.sjtu.edu.cn访问，用户名和密码与jaccount帐号一样。

如果现有portal上的数据与oldportal上的不同，请用户自己完成差异数据的导入，不便之处敬请原谅。

有些学生帐户的容量超过了200M，网络信息中心不能将这些数据导入到新设备上，请相关用户自行完成数据的导入。

上海交通大学网络信息中心

2006年11月3日

ata\_len=inversion(data\_len)

data\_len=int(data\_len,2) #二进制字符串中包含文档字节数和赫夫曼字典的位数

binary=source[24:24+data\_len] #记录文本字节数和编码字典的二进制字符串

code\_dict,text\_len=get\_data(binary)

decode\_str=source[24+data\_len:] #对原文档编码的字符串

a=list(code\_dict.keys()) #码字

num=[]

len1=inversion(len)

len1=int(len1,2)

print("encoded file bytes:",len1)

j=0

b=''

while(text\_len>0): #记录已经译码的字节数，直至已译码的字节数=原文档字节数

b=b+decode\_str[j]

for i in range (0,a.\_\_len\_\_()):

if(a[i]==b): #找出码字

num.append(b)

text\_len=text\_len-1

b=''

break

j=j+1

string='' #对各码字译码

for i in num:

string+=code\_dict[i]

str\_int=int(string,2)

str\_hex=hex(str\_int)

str\_hex=str\_hex.lstrip('0x')

code\_byte=bytes.fromhex(str\_hex)

with open(res\_file,'wb')as f: #解码后的字符串二进制写回

f.write(code\_byte)

print("decoded file bytes:",code\_by

《来自新世界》作者：[日]贵志祐介

编辑推荐

　　“新的秩序，要经过无数鲜血浇灌之后才能诞生。”

　　被网友们评为“恶意毕现”到令人想要膜拜的奇幻恐怖小说“神作”。

　　“鬼才”作家贵志祐介的科幻反乌托邦作品。

　　网络上受到热捧的同名动画片《来自新世界》原著。

　　第29回日本科幻小说大奖获奖作品。

　　关于1000年以后“未来世界”的奇幻故事。

内容简介

　　故事发生于1000年以后，人们放弃了高科技，过着日出而作、日落而息的前现代农业生活。与人类的祖先不同的是，这个社会里，每个人都有意念移物、意念点火（“咒力”）等超能力。拥有凡人一样的躯体，却有神一样的能力，有着如此矛盾结构的新人类组成的世界，称为“新世界”。然而这种神的能力对人来说，究竟意味着什么呢？

　　轻而易举就能改变物质，告别了一切机械和科技依赖，和大自然水乳交融。学校教育也只有两个等级，幼儿园和“咒力”学校。高强度的劳动由一种新出现的智力生物——化鼠——承担。人人遵守着代替法律的伦理规定，人和人之间洋溢着爱的氛围，生活天然就是幸福的。然而假象看上去有多美，揭开假象之后看到的现实就有多可怕……

　　“离开小町回头去看，有一件事我看得清清楚楚——我们的小町，很扭曲。

　　为了维持小町的安定和秩序，不断杀害孩子们的小町，还能称之为人类的社会吗？人类从涂满鲜血的历史中走过，才抵达如今的状态。然而即使是和过去黑暗的时代相比，今天的小町也是并不值得自豪的替代品。

　　我们已经无法在神栖六十六町活下去了。小町不许我们活下去。只要被打上了失格的烙印，便无法再回到当初了。这不是对待人类的方式，而是和甄选不良品一样的做法，你不觉得吗？当烧瓷窖开启的时候，走型的、有裂纹的瓷器，等待它们的就是被敲碎的命运。”

作者简介

　　贵志祐介，1959年生于大阪，毕业于京都大学。30岁时因为同事的意外死亡而改变人生轨迹，从公司辞职，专职从事写作。擅长推理、恐怖和科幻小说，处女作《第十三种人格》，代表作《黑屋吊影》、《青之焰》、《恶之教典》等。获奖记录：日本恐怖小说大奖（1997），日本推理作家协会奖（2005），日本SF大奖（2008），山田风太郎奖（2010）。

序

作者序

给台湾读者序

贵志祐介

　　我第一次拜访台湾是在十年前。当时为了纪念《玻璃之锤》上市，在微风广场举办簽名会，比想像中更多的读者莅临现场，温暖欢迎我到来，这幅景象至今深深留在我的心中。

　　在夜市大啖鸭舌，对臭豆腐退避三舍，参观关帝庙、故宫博物院、台湾大学，在地下街接受占卜师的占卜，给中医师把脉，还有体验让人叫苦连天的脚底按摩。明明是初次造访台北，但不可思议地有一股怀念，不禁让人浮现再次造访这座城市的念头。

　　这次，《来自新世界》在充满回忆的台湾翻译出版，我非常高兴，也十分荣幸。

　　这部作品的故事舞台是发生在离现今千年后的日本小镇，当时的人类获得了光靠意念就能移动物体的能力，而尽管世界表面上看起来是桃源乡，但在无法抑制好奇心和行动力的孩子眼中，逐渐映照出无比恐怖的真相。请沉醉在故事中，想像台湾发生同样的事情，如此一来，阖上书的时候，每个人的心中都会出现对人类和社会的崭新见解。

推荐序

我相信「神栖66町」真实存在著──阅读《来自新世界》

陈夏民（逗点文创结社总编辑）

　　原本就擅长用小说揭穿人性黑暗面的惊悚小说家贵志祐介，重回科幻创作跑道，交出一部质量超重量级的小说《来自新世界》，他经由对心理学的精通研究，辅以人物、念能力、社会、礼仪、法制、奇幻生物、世界观等钜细靡遗的设定，打造出一座「美得有病」的乌托邦城镇「神栖66町」，再让少年主人翁们逐渐发觉美好生活背后的丑陋真相，并在书末大屠杀的悲鸣中，上演一场血流成河、惊天动地的革命灾祸。

　　这部世界观庞大无比的作品，读来却没过分负担和沉重，可见贵志祐介说故事的能力无从挑剔。《来自新世界》内许多特別设定不落传统奇幻/科幻小说的窠臼，亦带弦外之音，充满奇趣。其中，仿佛出没身边的怪物、非典型少年英雄更是笔者最喜爱、折服的部分。

　　仿佛出没身边的怪物

　　恶鬼的个子与少年差不多，但长相无比狰狞，他仿佛要烧尽一切的愤怒，形成烈焰般的背光，汹湧不停地旋转。恶鬼所经之处，草木接连枯萎倒下，接着开始爆炸，燃起熊熊火焰。

　　少年脸色铁青，却忍著不敢尖叫，静静后退。钻过注连绳进入八丁标，恶鬼应该就看不见他了。但此时少年踩断枯枝，发出劈啪一响。

　　恶鬼面无表情地转头望向少年，仿佛终于找到发洩怒气的对象，紧盯他不放。

　　少年穿过注连绳，拔腿就逃。进入八丁标中就没事了。

　　没想到回头一看，恶鬼也钻过注连绳追上来！

　　少年这才发现自己犯下无可挽回的滔天大错，将恶鬼带进八丁标之中。

　　少年哭着在山路上狂奔，恶鬼在身后紧追不舍。