《多媒体技术》实验报告 6

黄勖 22920212204392

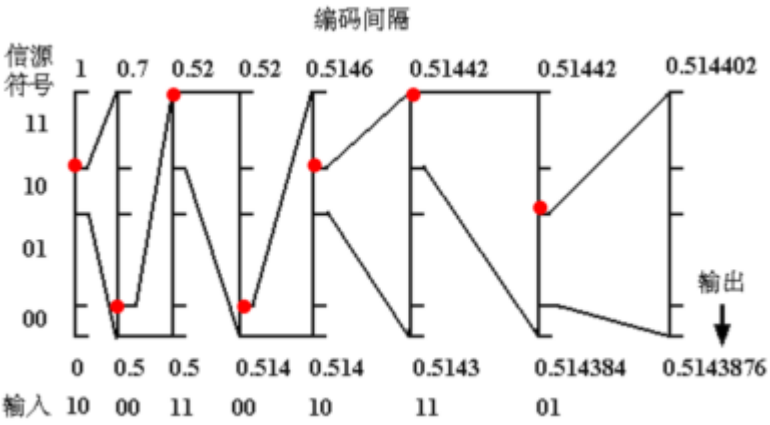
1.运行程序截图和简要说明

复习算术编码算法，基本步骤如下：

1.扫描整个文本文件，统计文件中每个字符出现的频率

2.每个字符将区间[0,1)分割成若干子间隔

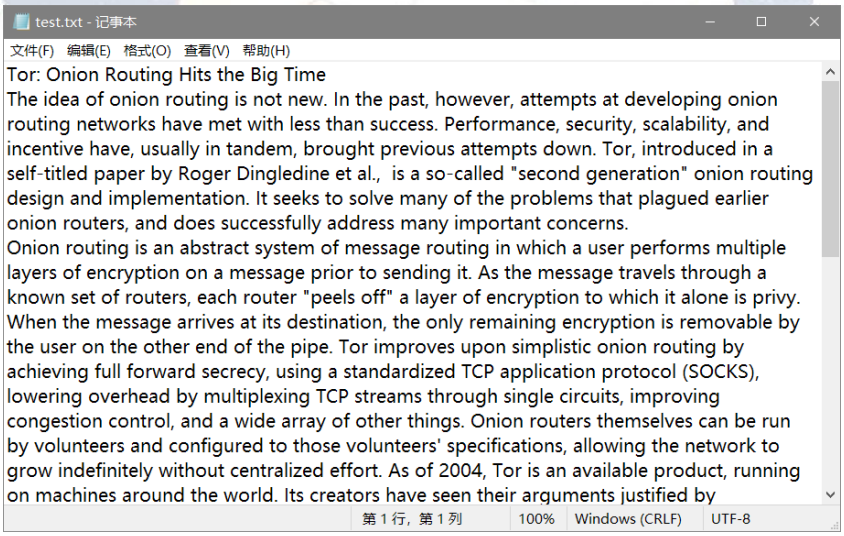
3.将文本文件中的字符串映射到[0,1)上某个子区间上



4.从子区间任选一个小数，并转为二进制，输出到目标文件

编码结果：

1）利用之前实验的文本数据当作编码元数据



2）编码字符串数据

```
# origin_txt 的内容必须出自统计字符样本文件test.txt
origin_txt = "gotointer"
with open("test.txt", "r", encoding="gbk") as text:
    ori = text.readline() # 读取样本文件
    p_space = get_probability(ori) # 计算概率空间
    print("样本文件中字符集为： ", p_space)
    interval = arithmetic_encode(p_space, origin_txt) # 对 origin_txt 进行编码
decoding = arithmetic_decode(p_space, interval, len(origin_txt))
print(origin_txt + " 对文章符号编码结果为： ", interval)
print("对文章符号译码结果为： ", decoding)
```

```
问题  输出  终端  调试控制台                                    Python Debug Console

PS E:\大二下\多媒体技术\lab6>  e:; cd 'e:\大二下\多媒体技术\lab6'; & 'D:\Program Files\python.exe' 'c:\Users\ASUS\.vscode\extensions\ms-python.python-2023.8.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '57981' '--' 'E:\大二下\多媒体技术\lab6\main.py'
样本文件中字符集为：  {'g': 0.05405405405405406, 'u': 0.02702702702702703, 's': 0.02702702702702703, 'r': 0.02702702702702703, 'o': 0.08108108108108109, '\n': 0.02702702702702703, 'n': 0.08108108108108109, 'H': 0.02702702702702703, 'e': 0.05405405405405406, 'h': 0.02702702702702703, 'T': 0.05405405405405406, 'D': 0.02702702702702703, 'i': 0.13513513513513514, 'm': 0.02702702702702703, ' ': 0.16216216216216217, 't': 0.08108108108108109, ':': 0.02702702702702703, 'R': 0.02702702702702703, 'O': 0.02702702702702703}
gotointer 对文章符号编码结果为：  [0.01104068306281903, 0.011040683100214652]
对文章符号译码结果为：  ['g', 'o', 't', 'o', 'i', 'n', 't', 'e', 'r']
PS E:\大二下\多媒体技术\lab6>
```

```
gotointer 对文章符号编码结果为：  [0.01104068306281903, 0.011040683100214652]
对文章符号译码结果为：  ['g', 'o', 't', 'o', 'i', 'n', 't', 'e', 'r']
```

2.主要代码展示和分析

扫描整个文本，统计每个字符出现的频率，每个字符将并确定区间[0,1)上各个字符的子间隔（建立映射）

```
def get_probability(char_li):
    p_space = {}
    chars = set(char_li)
    len_char_li = len(char_li)
    for char in chars:
        times = char_li.count(char) # 统计字符出现次数
        p_space.update({char: times/len_char_li}) # 计算概率
    return p_space
```

编码

```
def arithmetic_encode(p_space, origin_code): # origin_code 为待编码的字符串
    code = 0
    arit = 1
    keys = list(p_space.keys()) # 字符集
    for char in origin_code: # 对每个字符进行编码
        index = keys.index(char) # 字符在字符集中的位置
        pd = 0
        for i in range(index): # 计算累加概率
            pd = pd + p_space[keys[i]]
        code = code + pd * arit # 累加概率乘以区间长度
        arit = p_space[char] * arit # 区间长度缩小
        # print(char, code, arit)
    interval = [code, code + arit] # 返回编码区间
    return interval
```

解码

```python
def arithmetic_decode(p_space, interval, l): # l 为待译码的字符长度
    decoding_code = [] # 译码结果
    arit_range = {} # 每个字符对应的区间
    c = 0
    for char in p_space.keys():
        a = p_space[char]
        arit_range.update({char: (c, c+a)}) # 计算每个字符对应的区间
        c += a
    code = random.uniform(interval[0], interval[1])
    ori_range = arit_range.copy()
    while len(decoding_code) < l:
        for char, rang in ori_range.items():
            if rang[0] < code < rang[1]:
                decoding_code.append(char)
                c = rang[0]
                for ch, ran in arit_range.items():
                    a = ran[1] - ran[0]
                    ori_range.update({ch: (c, c + a * (rang[1] - rang[0]))})
                    c = c + a * (rang[1] - rang[0])
                break
    return decoding_code
```

主函数定义

字符串 origin_txt 用于计算转换小数、文档 test.txt，用于统计字符出现频率

```python
# origin_txt 的内容必须出自统计字符样本文件test.txt
origin_txt = "gotointer"
with open("test.txt", "r", encoding="gbk") as text:
    ori = text.readline() # 读取样本文件
    p_space = get_probability(ori) # 计算概率空间
    print("样本文件中字符集为： ", p_space)
    interval = arithmetic_encode(p_space, origin_txt) # 对 origin_txt 进行编码
decoding = arithmetic_decode(p_space, interval, len(origin_txt))
print(origin_txt + " 对文章符号编码结果为： ", interval)
print("对文章符号译码结果为： ", decoding)
```

编解码结果

```
gotointer 对文章符号编码结果为：  [0.01104068306281903, 0.011040683100214652]
对文章符号译码结果为：  ['g', 'o', 't', 'o', 'i', 'n', 't', 'e', 'r']
```

3.其他

这次试验我实际编写了区间编解码程序，通过实践体会了编码与解码的实际运行过程，收获颇丰。