

## Assignment #3 <開源程式的使用> Result Report

姓名:蔡雨蓁 學號:108501537 系級:電機 4B

### 1. 編譯結果

```
tsai@LAPTOP-HJB975Q4:~/projects/assignment_3-bonnietsai/src$ g++ hw4_main.cpp main
```

### 2. 執行結果

```
tsai@LAPTOP-HJB975Q4:~/projects/assignment_3-bonnietsai/src$ ./a.out
```

#### ● 測資參數設定

- 測試程式之測資={ 'a' , 'b' , 'c' , 'd' , 'e' }
- 頻率分別假設為={ 0.05, 0.1 , 0.2 , 0.25 , 0.4 }
- 字母出現次數為={ 5 , 10 , 20 , 25 , 40 }
- 以字母總數量 100 為假設

#### ● Huffman code 編碼原理解釋:

- Huffman code 使用變長編碼表進行編碼，通過評估來源符號出現機率的方法得到，出現機率高的字母使用較短的編碼，反之出現機率低的則使用較長的編碼，這便使編碼之後的字串的平均長度、期望值降低，從而達到無失真壓縮數據的目的。
- 測試程式之結果分析:  
'a'之出現次數最低(5)，其編碼結果之 bit 數最長，符合其編碼原理。  
'e'之出現次數最高(40)，其編碼結果之 bit 數最短，符合其編碼原理。

```
-----Huffman-----:
e: 0
d: 10
a: 1100
b: 1101
c: 111
```

Huffman code 編碼結果

#### ● Arithmetic coding 編碼原理解釋:

在給定符號集和符號概率的情況下，算術編碼可以給出接近最佳的編碼結果。使用算術編碼的壓縮演算法通常先要對輸入符號的概率進行估計，然後再編碼。這個估計越准，編碼結果就越接近最佳的結果。

```
-----arithmetic coding-----:
```

```
Enter probability of each character:
```

Symbol	Probability	Range_from	Range_to
a	0.05	0	0.05
b	0.1	0.05	0.15
c	0.2	0.15	0.35
d	0.25	0.35	0.6
e	0.4	0.6	1

Arithmetic coding  
區段編碼

➤ 測試程式之結果分析:

‘a’之出現機率最低(0.05)，其編碼後區間最短，符合其編碼原理。

‘e’之出現機率最高(0.4)，其編碼後區間最長，符合其編碼原理。

● Arithmetic coding 編碼解碼結果分析

➤ 可由下兩張圖發現 abccc 編碼結果為 0.00343，但 0.00343 解碼後卻變為 abccb，可看出兩結果不符合。

➤ 原因可能為編碼時小數位數的長度大於浮點數數可表示的範圍，損失了小數點較後面的位數，導致其失去精確度，造成解碼時之錯誤。

```
Enter text(only using a,b,c,d,e && length 5)ex:abcba : abccc
```

```
Encoding
```

Symbol	Low_v	High_v	diff
a	0	0.05	0.05
b	0.0025	0.0075	0.005
c	0.00325	0.00425	0.001
c	0.0034	0.0036	0.0002
c	0.00343	0.00347	4e-05

```
Code word for abccc is: 0.00343
```

Arithmetic coding  
abccc 編碼結果 0.00343

```
Decoding:
```

Code	Output	Range_from	Range_to
0.00343	a	0	0.05
0.0686	b	0.05	0.15
0.186	c	0.15	0.35
0.18	c	0.15	0.35
0.15	b	0.05	0.15

```
Text for 0.00343 is: abccb
```

Arithmetic coding  
0.00343 解碼結果為 abccb

- 兩種方式編碼時間比較

- 可由下列編譯結果發現 arithmetic coding 使用之時間較長，因為 Huffman 在編碼前須先建立二元樹，因此可發現編碼時間較久，而算數編碼僅需要使用出現機率劃分單位因此執行時間較短。

```

-----兩種方式的編碼各自的編碼時間-----:
進行運算所花費的時間(huffman) : 0.000114 S
進行運算所花費的時間(arithmetic coding) : 7.1e-05 S
  
```

編碼時間比較

- 兩種方式編碼特性比較

Arithmetic coding	Huffman coding
Arithmetic coding is not a statistical method	Huffman coding is a statical method
It yield an optium result	It does not yield an optium result
There is no one to one correspondence between source symbol and code word	There is one to one correspondence bwteewen source and code word
For eg. If a,b,c messages then only one unique code to entire message	For eg. If a,b,c are messages then separate symbols are assigned