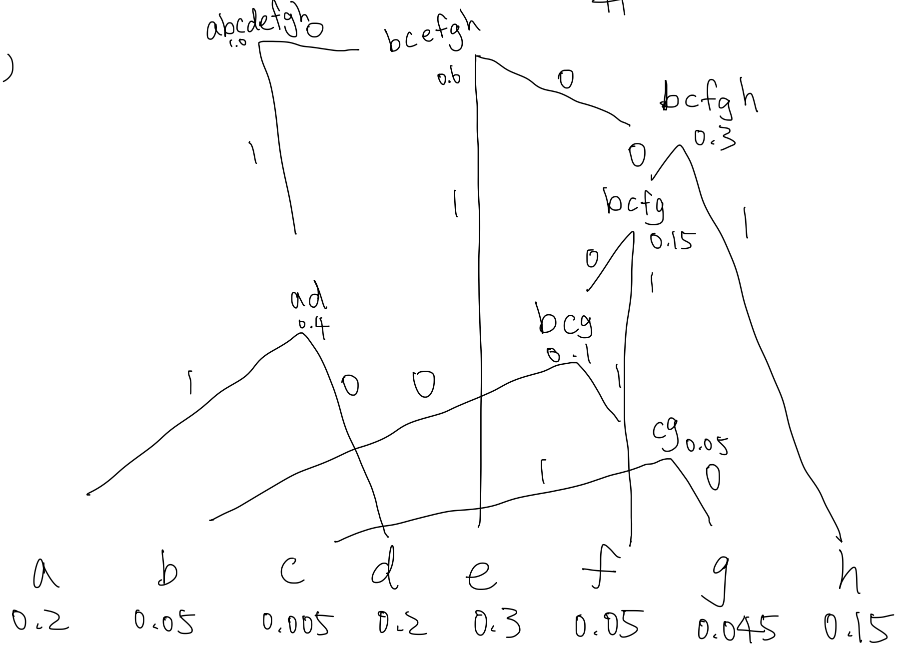
B10901184 錡亭勳 Lab4

1.

(a) $-\left(0.2\log 0.2 + 0.05\log 0.05 + 0.005\log 0.005 + 0.2\log 0.2 + 0.3\log 0.3 + 0.05\log 0.05 + 0.045\log 0.045 + 0.15\log 0.15\right)$

≈ 2.53214 bits (base 2) #

(b)



	prob.	left child	right child	bits
a	0.2			11
b	0.05			00000
c	0.005			000011
d	0.2			10
e	0.3			01
f	0.05			0001
g	0.045			000010
h	0.15			001
cg	0.05	g	c	(00001)
bcg	0.1	b	cg	(0000)
bcfg	0.15	bcg	f	(000)
bcfgh	0.3	bcfg	h	(00)
ad	0.4	d	a	(1)
bcefgh	0.6	bcfgh	e	(0)
abcdefgh	1.0	bcefgh	ad	

(c)

$2^{-2}+2^{-5}+2^{-6}+2^{-2}+2^{-2}+2^{-4}+2^{-6}+2^{-3}=1$

Satisfy #

(d) #

$0.2\times2+0.05\times5+0.005\times6+0.2\times2+0.3\times2+0.05\times4+0.045\times6+0.15\times3$

$=2.6$

by (a) $H[X]\approx2.53\leq2.6<H[X]+1$ satisfy

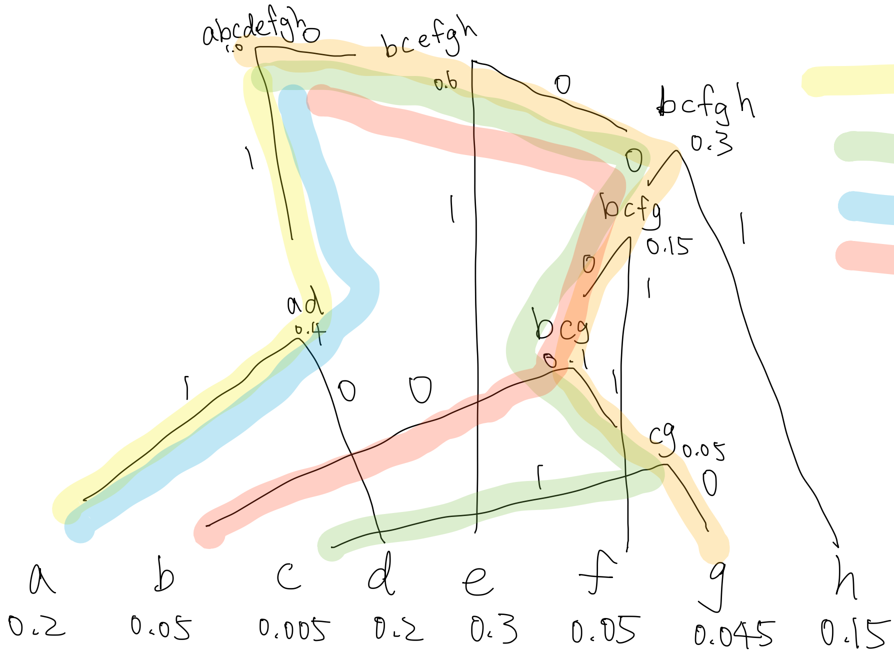
$\bar{L}=2.6$ #

(e)

$\{g, a, c, a, b\}=\{000010, 11, 000011, 11, 00000\}$

$\rightarrow$ 000010110000111100000 #

(f)



g 000010
a 11
c 000011
a 11
b 00000

→ gacab

#

(g)

find sequences s.t. $\left| \frac{-\log p(x^{10})}{10} - H[X] \right| < 0.1$

ahahahahah, haahaahhah, aaaaahhhhh

$\left| \frac{-\log(0.2 \times 0.15)^5}{10} - 2.53214 \right| \approx 0.00269 < 0.1$

hehhehhhhh, ehehhhhhhh, hheehhhhhh

$\left| \frac{-\log(0.15)^8 \times (0.3)^2}{10} - 2.53214 \right| \approx 0.00483 < 0.1$

beeebeebee, ebeebeebee, eebebeebee, eeeeeeebbb

$\left| \frac{-\log(0.05)^3 \times (0.3)^7}{10} - 2.53214 \right| \approx 0.01969 < 0.1$

#

B10901184 錡亭勳 Lab4

2.

(a)

```
dict =

  15×5 cell array

    {'a'        }    {[0.2000]}    {0×0 double}    {0×0 double}    {'11'      }
    {'b'        }    {[0.0500]}    {0×0 double}    {0×0 double}    {'00000'   }
    {'c'        }    {[0.0050]}    {0×0 double}    {0×0 double}    {'000011'  }
    {'d'        }    {[0.2000]}    {0×0 double}    {0×0 double}    {'10'      }
    {'e'        }    {[0.3000]}    {0×0 double}    {0×0 double}    {'01'      }
    {'f'        }    {[0.0500]}    {0×0 double}    {0×0 double}    {'0001'    }
    {'g'        }    {[0.0450]}    {0×0 double}    {0×0 double}    {'000010'  }
    {'h'        }    {[0.1500]}    {0×0 double}    {0×0 double}    {'001'     }
    {'gc'       }    {[0.0500]}    {[         7]}    {[         3]}    {'00001'   }
    {'bgc'      }    {[0.1000]}    {[         2]}    {[         9]}    {'0000'    }
    {'bgcf'     }    {[0.1500]}    {[        10]}    {[         6]}    {'000'     }
    {'bgcfh'    }    {[0.3000]}    {[        11]}    {[         8]}    {'00'      }
    {'da'       }    {[0.4000]}    {[         4]}    {[         1]}    {'1'       }
    {'bgcfhe'   }    {[0.6000]}    {[        12]}    {[         5]}    {'0'       }
    {'bgcfheda'}    {[      1]}    {[        14]}    {[        13]}    {0×0 double}
```

(b) same as Problem 1 (e).

```
>> bin_seq = huffman_enc(sym_seq, dict)

bin_seq =

    '0000101100001111000000'
```

(c) same as Problem 1 (f).

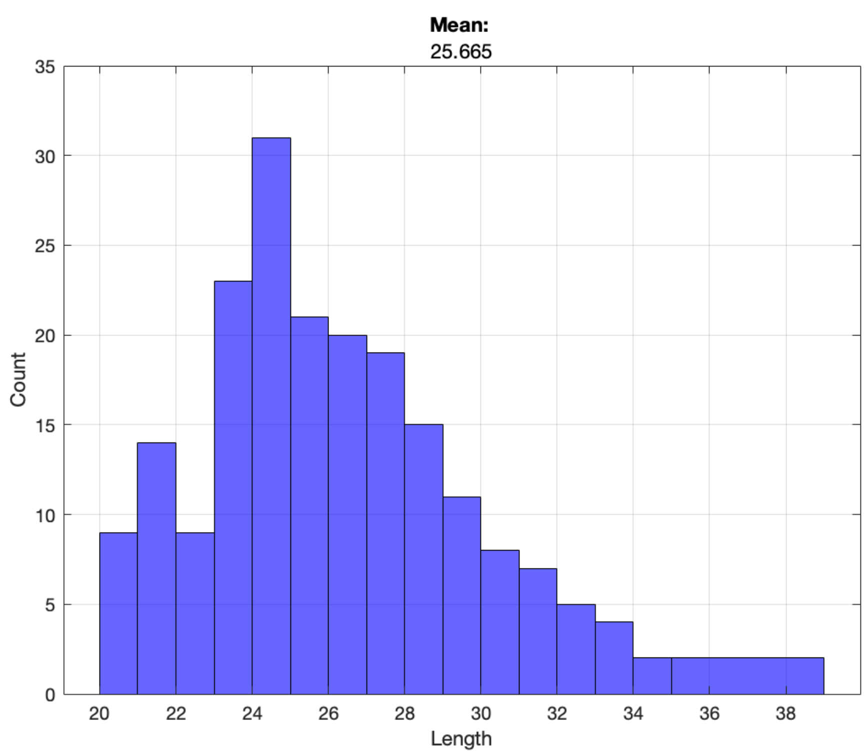
```
>> sym_seq = huffman_dec(bin_seq, dict)

sym_seq =

  1×5 cell array

    {'g'}    {'a'}    {'c'}    {'a'}    {'b'}
```

3.

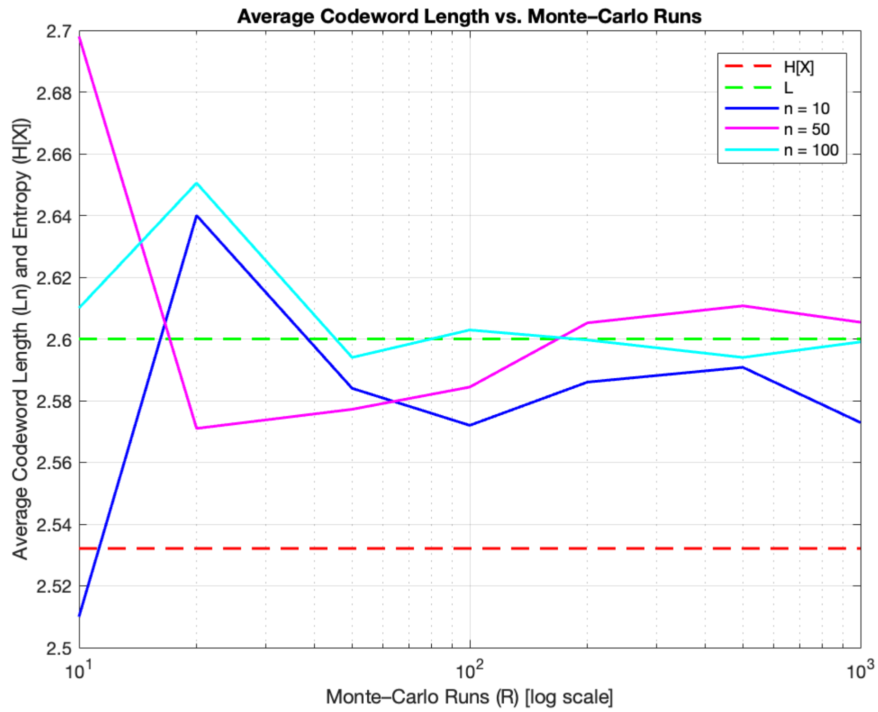
(a)

```
>> huffman_sample(10, 1, false, false)

symbols =

  1×10 cell array

  Columns 1 through 9

    {'h'}    {'a'}    {'e'}    {'d'}    {'e'}    {'d'}    {'d'}    {'e'}    {'a'}

  Column 10

    {'h'}


bin_seq =

    '0011101100110100111001'


len =

    22
```

(b)



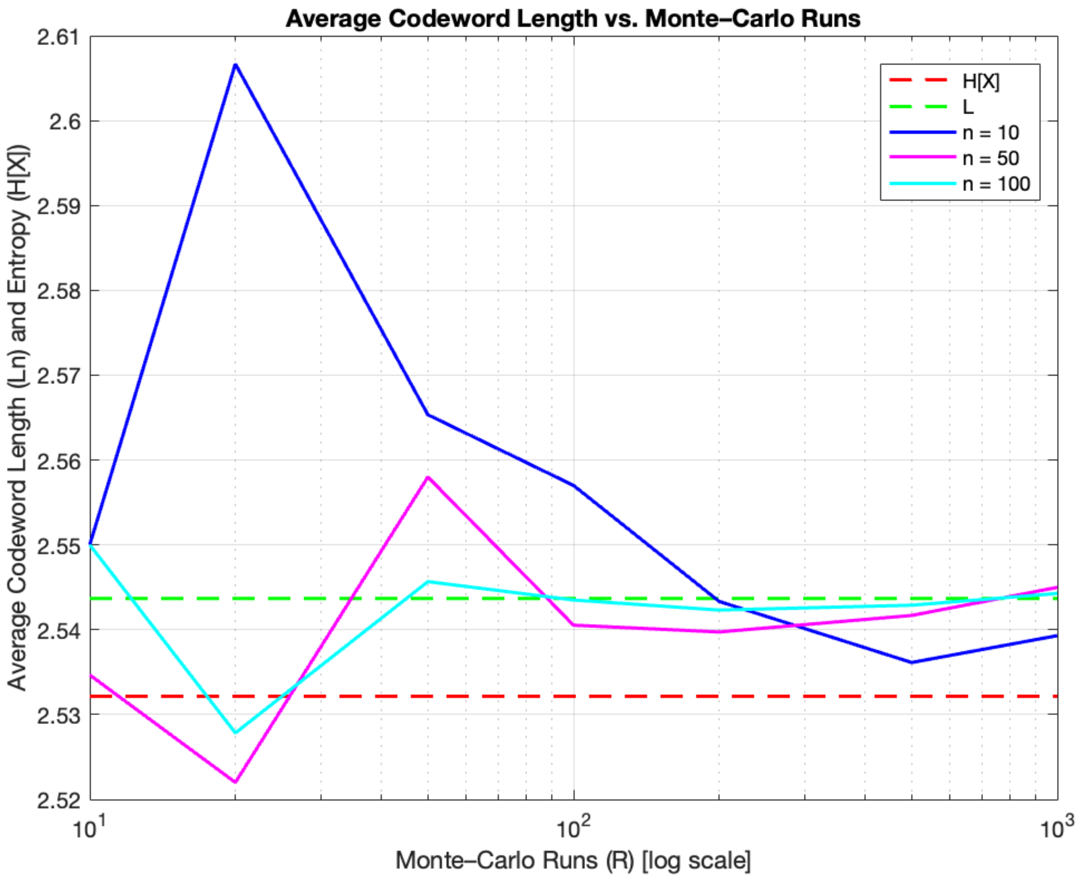
(c)



(d)

As the number of runs increases, the average of the average of length approximates the true mean. As the number of samples, the trends are also the same. This tells us that the length under the Monte–Carlo sample is valid when the number of sampling points and the number of runs approach infinity, which is good since when tend to transmit a lot of data.

4.



The difference between the entropy and the mean of the length of code decreases compared to only one system. This is the fact mentioned in class, the 1/N term that bounds the average L decreases as N increases.