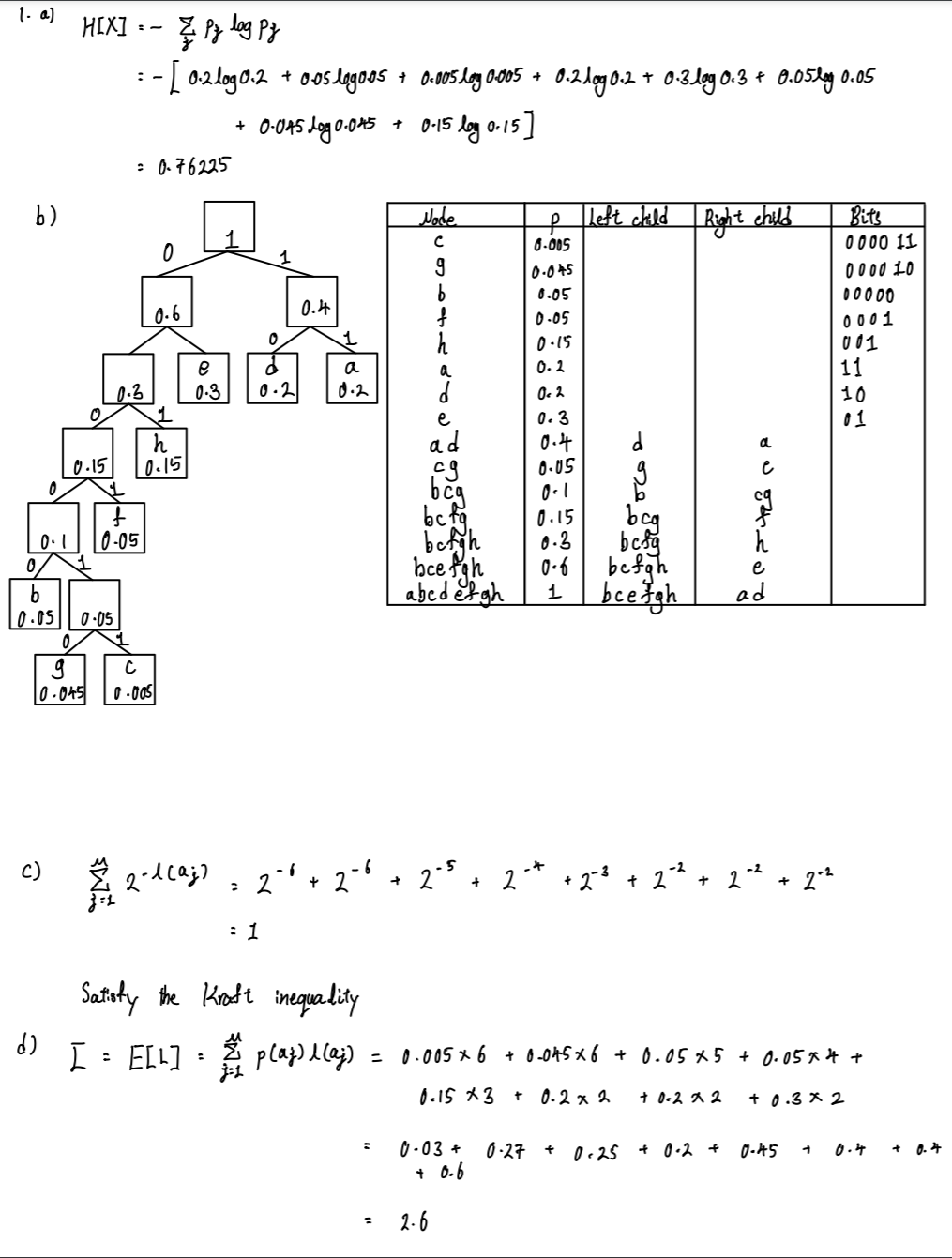
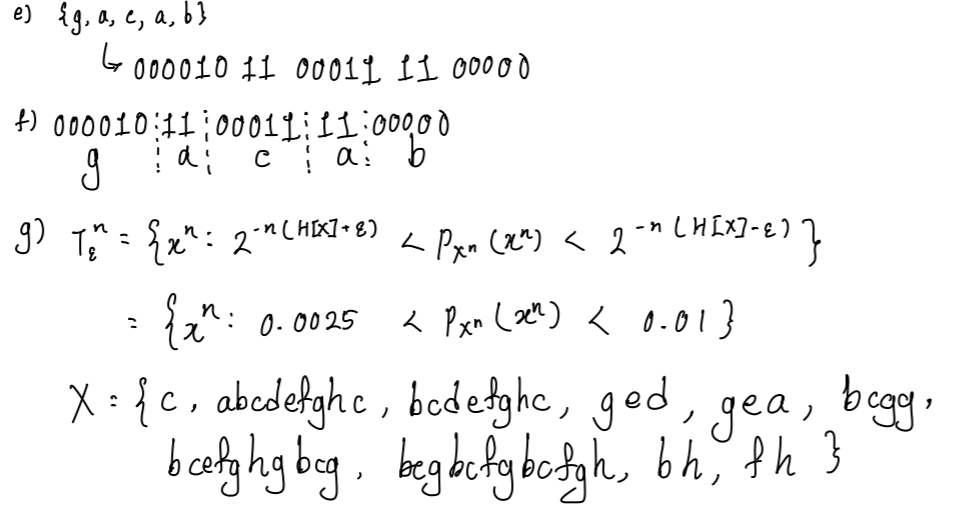
*陳志偉 B06901126電機五*

**Electrical Engineering Lab（topics on Communication System）**

**Lab4 Report**





2.a) Figure 1 show the dictionary of Table 1.

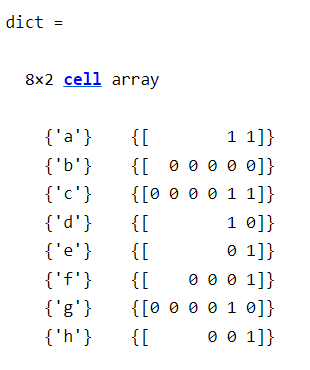


Figure 1

2.b) After encoded the sequence of symbols {g, a, c, a, b}, the bits strings showed as Figure 2.

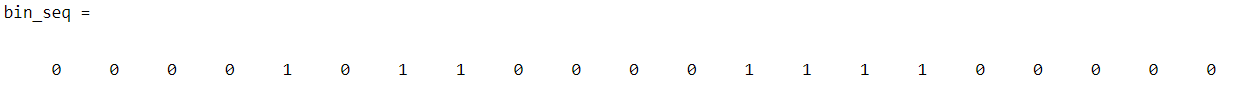


Figure 2

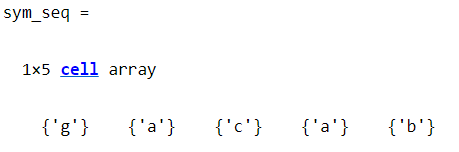
2.c) Let decode the bits strings from 2.b), we get the same result before we encoded the sequence of symbols {g, a, c, a, b}.

Figure 3

3.a) From Figure 4, ‘e’ has the highest frequency showing in the random string which is satisfied with the probability in Table 1 and the length of the random string is 23 which is also satisfied with dictionary showed as Figure 1.

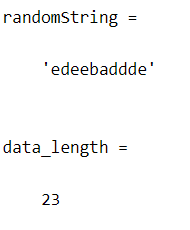


Figure 4

3.b)

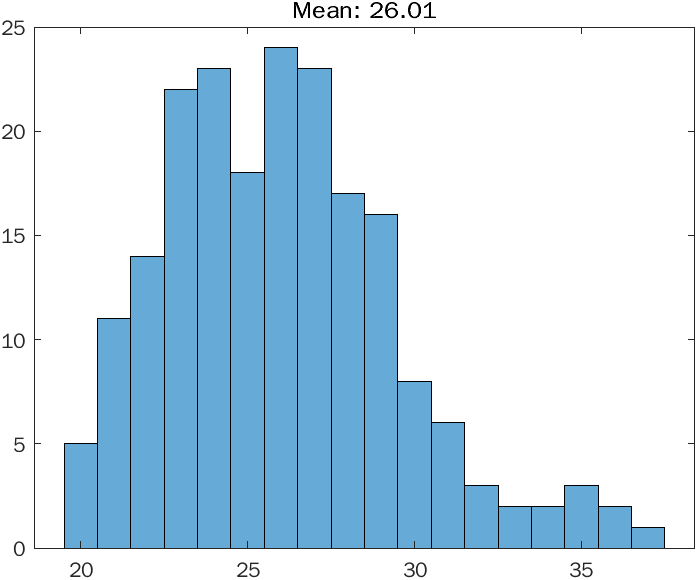


Figure 5

3.c)

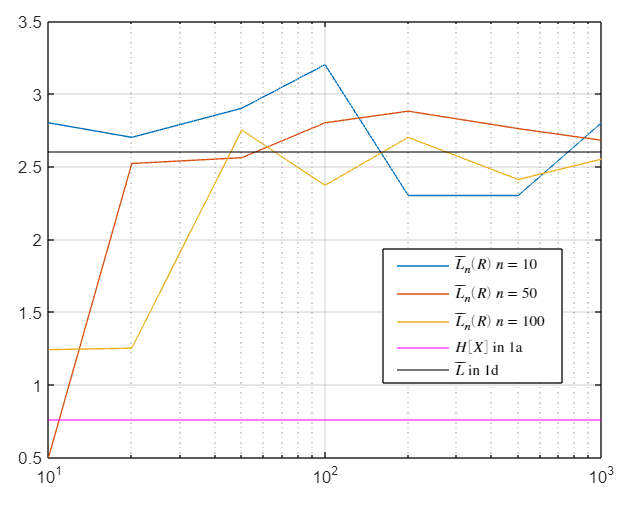


Figure 6

3.d) The three curves obtained in 3.c) will eventually converge to the answer we got in 1d, no matter how much N is equal to, as long as R is large enough. If N is large enough and R is small, the average codeword length will close to entropy.

***Appendix***

Code of problem 2

1. symbols = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
2. prob = [0.2, 0.05, 0.005, 0.2, 0.3, 0.05, 0.045, 0.15];
3. dict = huffmandict( symbols, prob );
4. display(dict);
5. sym\_seq = {'g', 'a', 'c', 'a', 'b'};
6. display(sym\_seq);
7. bin\_seq = huffmanenco(sym\_seq, dict);
8. display(bin\_seq);
9. sym\_seq = huffmandeco(bin\_seq, dict);
10. display(sym\_seq)

Code of problem 3

1. symbols = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
2. prob = [0.2, 0.05, 0.005, 0.2, 0.3, 0.05, 0.045, 0.15];
3. sym\_bit = {'11', '00000', '000011', '10', '01', '0001', '000010', '001'};
4. N = [10, 50, 100];
5. R = [10 20 50 100 200 500 1000];
6. data\_length\_array = [];
7. avg\_codeword = [];
8. for n = 1:length(N)
9. for k = 1:length(R)
10. for j = 1:R(k)
11. indices = randsrc(N(n),1,[1:numel(symbols); prob]);
12. randomString = [symbols{indices}];
13. %display(randomString);
14. data\_length = 0;
15. for i = 1:length(randomString)
16. idx = find(strcmp(symbols, randomString(i)));
17. data\_length = data\_length + length(sym\_bit{idx});
18. end
19. data\_length\_array(k, j) = data\_length;
20. end
21. mean = sum(data\_length\_array(n, k))/length(data\_length\_array(k));
22. avg\_codeword(n, k) = mean/N(n);
23. end
24. end
25. entropy\_1a = 0.76225;
26. avg\_codeword\_1d = 2.6;
27. display(avg\_codeword(1,:))
28. semilogx(entropy\_1a)
29. semilogx(R, avg\_codeword)
30. yline(entropy\_1a, Color='magenta')
31. yline(avg\_codeword\_1d, Color='black')
32. h = legend('$\overline{L}\_n(R)$ $n=10$', '$\overline{L}\_n(R)$ $n=50$', '$\overline{L}\_n(R)$ $n=100$', '$H[X]$ in 1a', '$\overline{L}$ in 1d','Interpreter','latex');
33. rect = [0.6, 0.25, 0.25, 0.25];
34. set(h, 'Position', rect)
35. grid on

*All code source will push to my github repo:* [*https://github.com/finalwee/CommLab*](https://github.com/finalwee/CommLab)