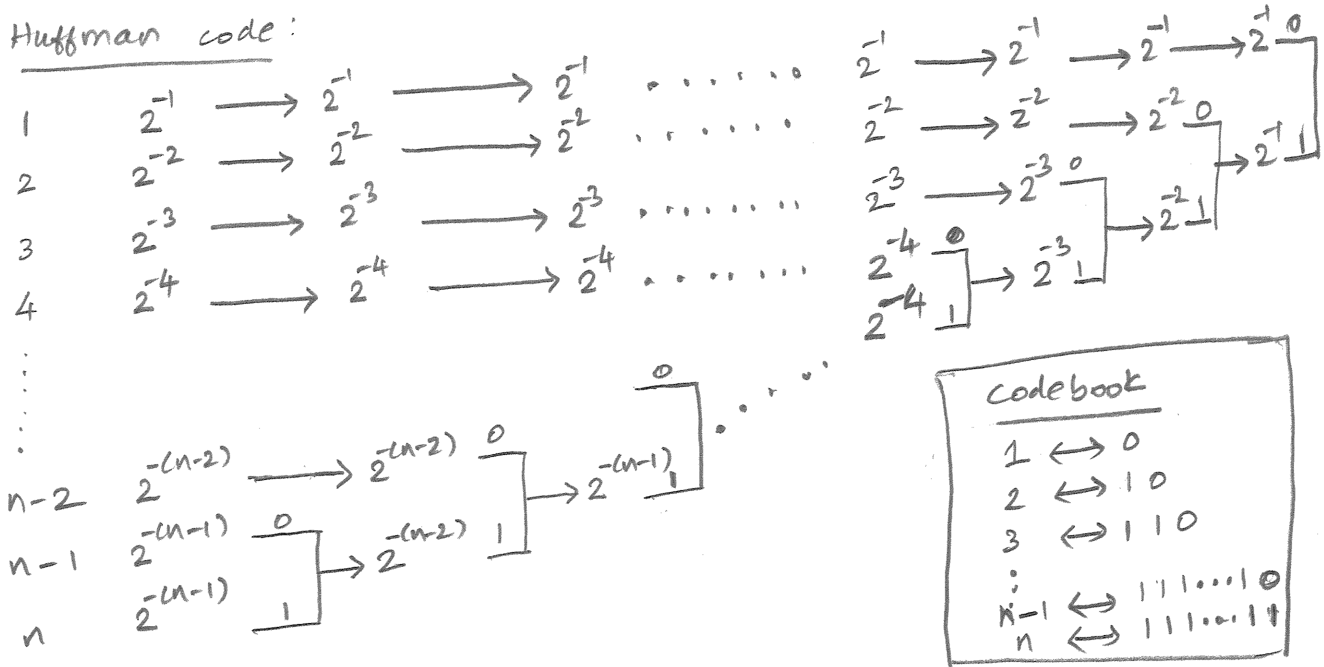
1 $X = (\mathcal{X}, p)$

$\mathcal{X} = \{1, \ldots, n\}$

$p(x) = 2^{-x}, \quad x = 1, \ldots, n-1, \quad p(n) = 2^{-(n-1)}$

Huffman code:



Codebook

$1 \leftrightarrow 0$
$2 \leftrightarrow 10$
$3 \leftrightarrow 110$
$\vdots$
$n-1 \leftrightarrow 111\cdots10$
$n \leftrightarrow 111\cdots11$

Observe that: $l(1) = 1,\ l(2) = 2,\ l(3) = 3 \ldots,\ l(n-1) = n-1.$
$l(n) = n-1$

$\therefore$ avg. codeword length $L(C,p) = \sum_{i=1}^{n-1} i \cdot 2^{-i} + (n-1)2^{-(n-1)}$ — ①

On the other hand,

the entropy $H(p) = -\sum_{i=1}^{n-1} 2^{-i} \log_2 2^{-i} - 2^{-(n-1)} \log_2 2^{-(n-1)}$

$= \sum_{i=1}^{n} i \cdot 2^{-i} + (n-1)\, 2^{-(n-1)}$

$= L(C,p)$ (From ①).

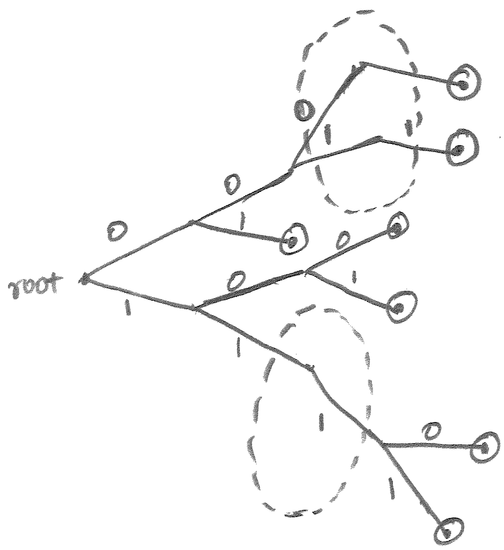
$\therefore$ the entropy bound is indeed achieved!

2.

Codebook:

$\{01, 100, 101, 1110, 1111, 0011, 0001\}$

Tree representation:



Observe that the codewords are not optimally assigned in the highlighted regions of the tree.

Hence, this cannot be a Huffman code.

3
9.9

Let us first take the case $K=3$.

Obviously, 3 symbols can not be represented by just 1 bit.

Thus, any fixed-length code has an average codeword length of at least 2 bits.

On the other hand, consider the variable-length code $\{0, 10, 11\}$. The average codeword length is

$$\frac{1+2+2}{3} = 1.67 \text{ bits}$$

and hence is more efficient than the fixed-length code.

Thus, the assertion is not true in this case.

Next, take K to be power of 2, i.e.

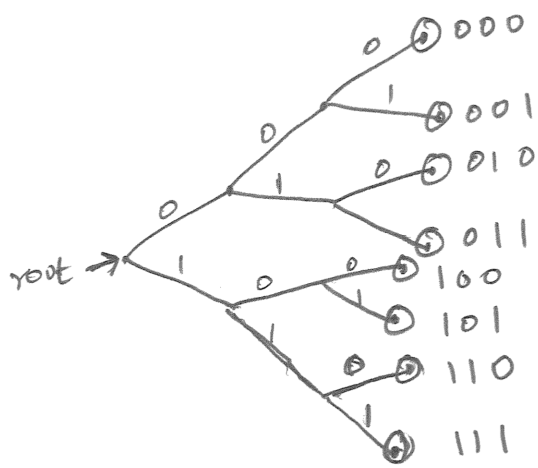
$$K = 2^n \quad , \quad n > 0 \text{ integer.}$$

Since the source is equiprobable, the optimal code have average codeword length

$$L(C,p) \geq H_2(X) = \log_2 K = \underline{n \text{ bits}}.$$

We will now construct a <u>fixed-length</u> code that has length n bits and hence is as efficient as any code can be.

We take the binary tree of <u>depth n</u> and label each of the 2^n leaves as the codewords. For e.g. whe $n = 3$,



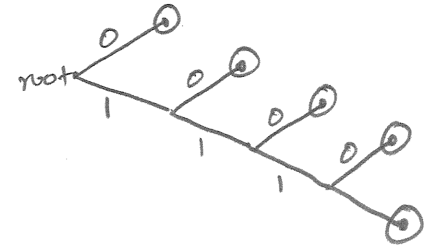
This <u>fixed-length</u> code is 100% efficient.

Thus, the assertion in the problem is true if

$$\underline{\underline{K = 2^n}} \quad , \quad n > 0 \text{ integer.}$$

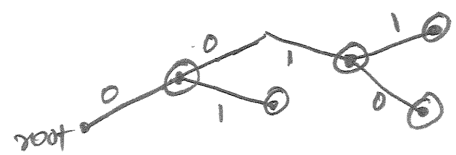
4
9.10

(a) code I:



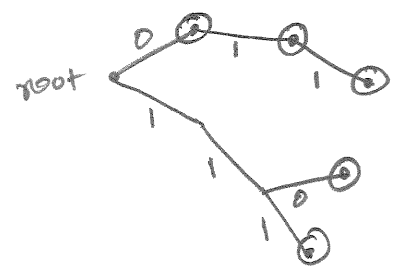
PREFIX !

code II:



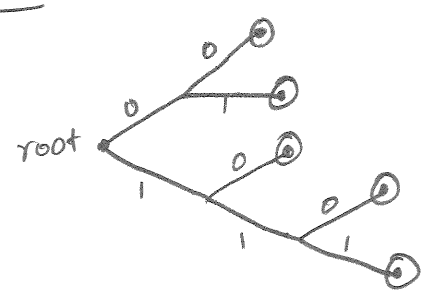
Not prefix.

code III:



Not prefix

code IV:



PREFIX !

(b) Kraft inequality:

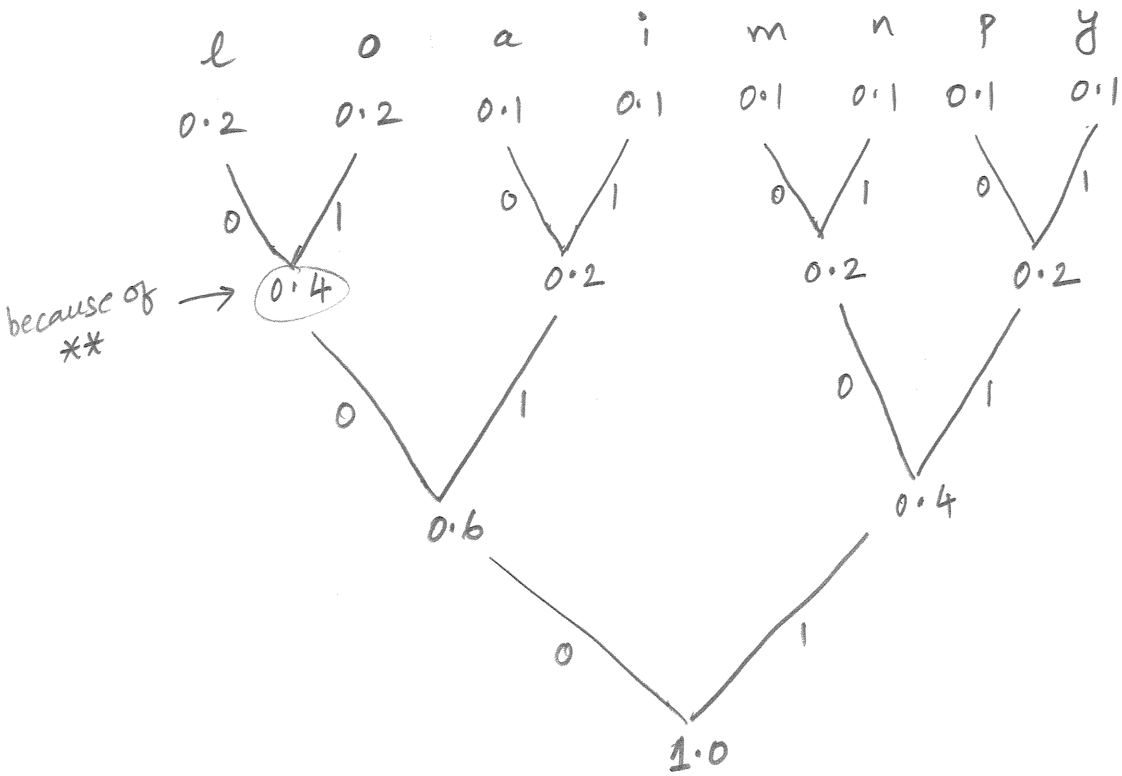
code I: $2^{-1} + 2^{-2} + 2^{-3} + (2 \times 2^{-4}) = 1$ satisfied !

code IV: $(3 \times 2^{-2}) + (2 \times 2^{-3}) = 1$ satisfied !

Since code I and code IV are prefix, Kraft inequality is satisfied as expected.

Huffman code 1 : (more combined symbol as high as possible).

* Here, a different way of representing the combining of symbols is demonstrated. You may choose to use either the one in the book, or this, whichever you find convenient.



** moving a combined symbol as high as possible essentially means combined symbols take the lowest precedence in combining further (when there are ties).

codebook:

a : 010

i : 011

l : 000

m : 100

n : 101

o : 001

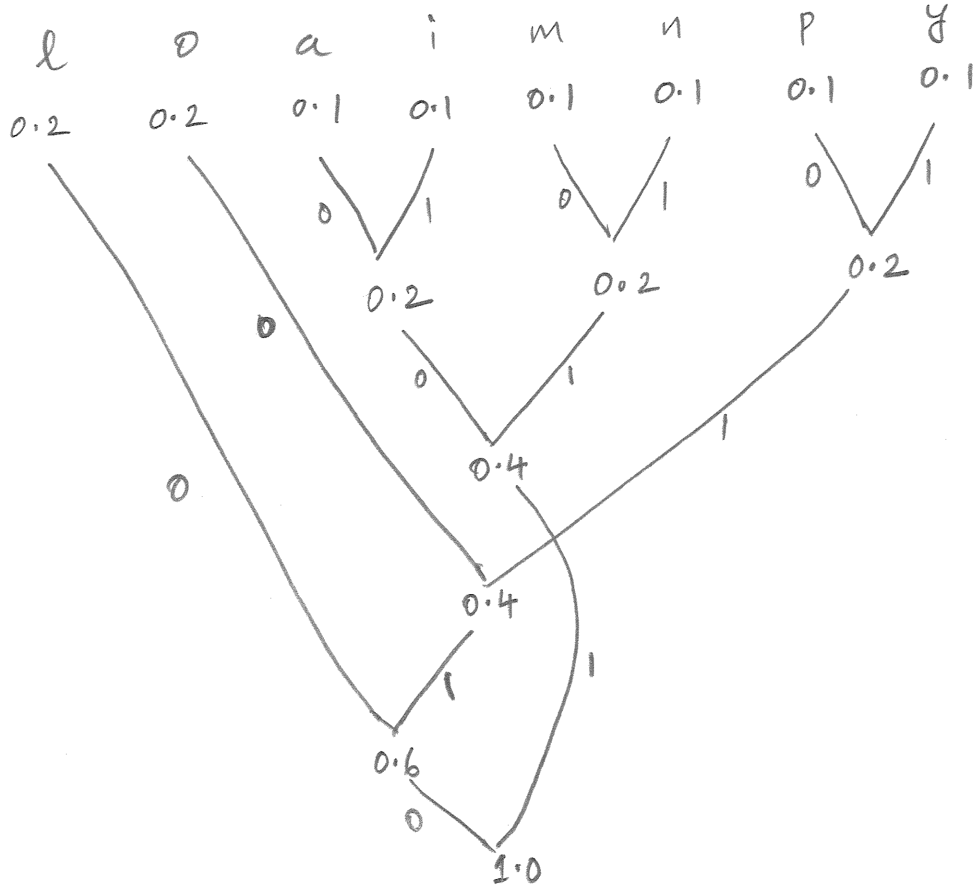
p : 110

y : 111

Avg. codeword length = 3

Variance = 0

Huffman Code 2 : (move combined symbol as low as possible i.e. combined symbols take highest precedence in combining further).



Codebook

a		100
i	:	101
l	:	00
m	:	110
n	:	111
o	:	010
p	:	0110
y	:	0111

Avg. codeword length =

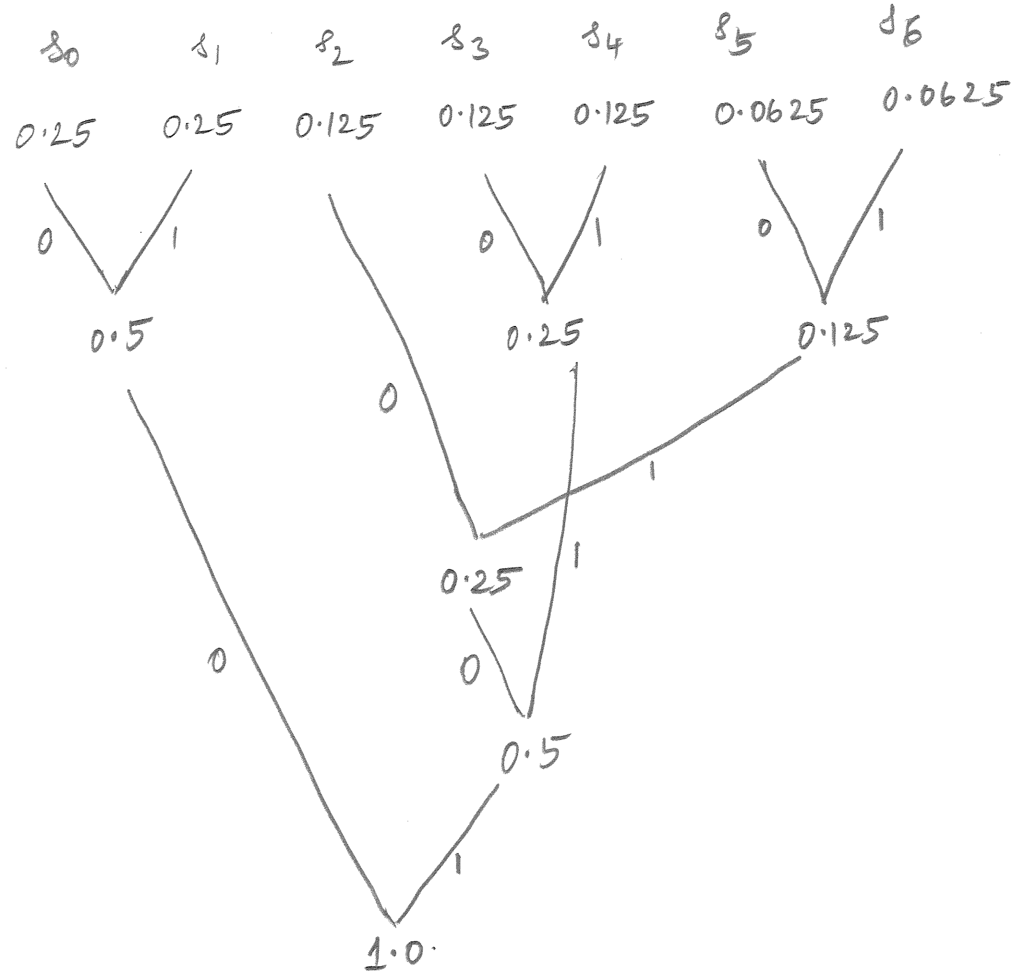
$0.2 \times (2 + 3) + 0.1 \times (4+4+3+3+3+3)$

$= 3$

Variance $= ((4-3)^2 \times 0.2) + ((2-3)^2 \times 0.2)$

$= 0.4$

$\frac{6}{9.12}$



Codebook:

s_0 : 00

s_1 : 01

s_2 : 100

s_3 : 110

s_4 : 111

s_5 : 1010

s_6 : 1011

Avg. codeword length =

$$(2 \times 0.25) \times 2 + (3 \times 0.125) \times 3 + (4 \times 0.0625) \times 2$$

$$= 2.625$$

$$H_2(S) = -\left(0.25 \log_2 \overset{=2^{-2}}{0.25}\right) \times 2 - \left(0.125 \log_2 \overset{=2^{-3}}{0.125}\right) \times 3 - \left(0.0625 \log_2 \overset{=2^{-4}}{0.0625}\right) \times 2$$

$$= (0.25 \times 2) \times 2 + (0.125 \times 3) \times 3 + (0.0625 \times 4) \times 2$$

$$= 2.625.$$

Thus, we have 100% efficiency, i.e. entropy bound is achieved.

This is because the pmf of source is of the form $\boxed{p(s_i) = 2^{-n_i}}$, with $n_i \in \mathbb{N}_0$, $i = 0, 1, \cdots, 6$.

7
9.14

Codebook:

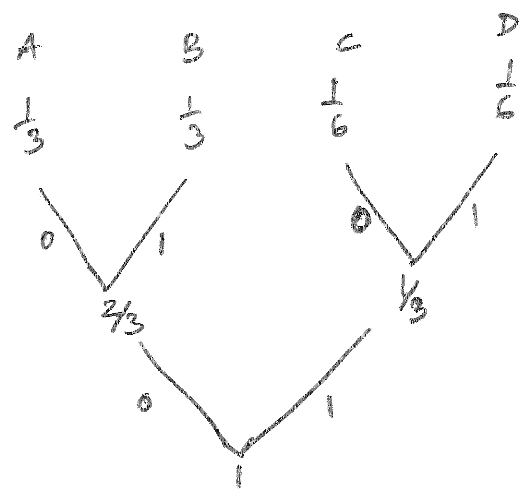
Symbol	codeword	length
A	1	1
B	011	3
C	010	3
D	001	3
E	0001	4
F	00001	5
G	00000	5

8

$\mathcal{X} = \{A, B, C, D\}$

Take $p = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6}\right)$

①



Codebook:

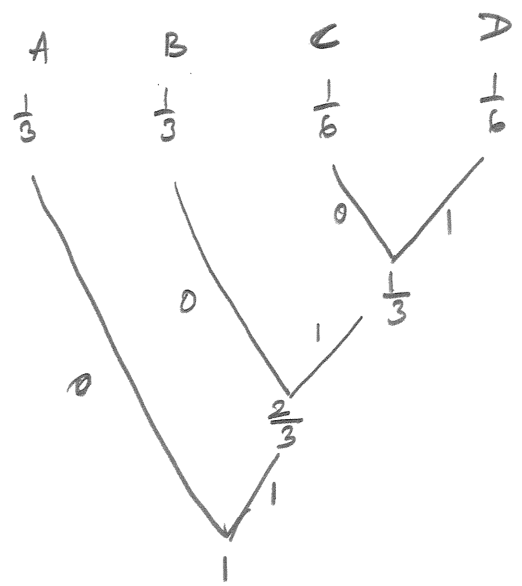
A : 00

B : 01

C : 10

D : 11

②



Codebook:

A : 0

B : 10

C : 110

D : 111

9

$X = (\mathcal{X}, \underline{p})$

Recall that the expected codeword length of the optimal code satisfies

$$H_2(X) \le L(C, \underline{p}) \le H_2(X) + 1.$$

Hence, this is also true for the Huffman code.

The difference satisfies

$$0 \le L(C, \underline{p}) - H_2(X) \le 1.$$

Thus, we will now try to find a source for which the above difference is close to 1.

Let us take a 2-symbol source, say $\mathcal{X} = \{0, 1\}$.

let $P(X=0) = 1-p$, $P(X=1) = p$. $(0 < p < \frac{1}{2})$.

$\therefore H_2(X) = -p \log_2 p - (1-p) \log_2 (1-p)$.

The optimal Huffman code uses 1 bit for each symbol and hence $L(C, p) = 1$.

In order to make the difference $L(C, p) - H_2(X)$ close to 1, we make the entropy $H_2(X)$ close to 0 by choosing p to be a small positive constant.

By taking p to be arbitrarily close 0, we can make the difference $L(C, p) - H_2(X)$ to be arbitrarily close to 1.

This gives the required source!

10
9.16

already stored

0 1 ¦ 1 1 | 1 0 | 1 0 0 | 1 1 0 | 0 0 | 1 0 1 | 1 0 1 0 | 0 ...

1 2 ¦ 3 4 5 6 7 8 9

not enough information.

Compressed: $\underbrace{0010}_{\text{4 bits for location}}1$ 00100 01000 00110 00010 01001 10000