

18 Arithmetic Coding

One of the most powerful compression techniques is called *Arithmetic Coding*. This converts the entire input data into a single floating point number. (A *floating point* number is similar to a number with a decimal point, like 3.5 instead of $3\frac{1}{2}$. However, in arithmetic coding we are not dealing with decimal numbers so we call it a floating point instead of a decimal point.)

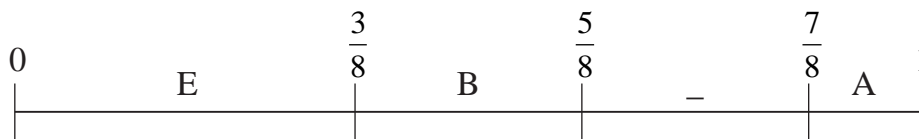
We will use as our example the string (or message)

BE_A_BEE

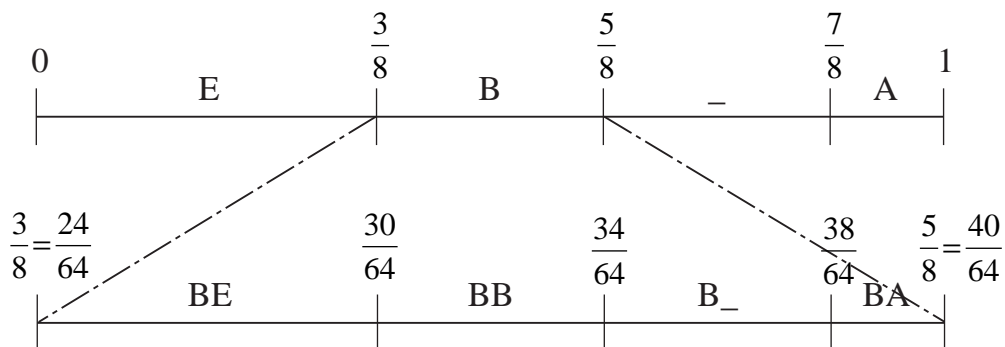
and compress it using arithmetic coding. The first thing we do is look at the frequency counts for the different letters:

E	B	_	A
3	2	2	1

Then we encode the string by dividing up the interval $[0, 1]$ and allocate each letter an interval whose size depends on how often it occurs in the string.

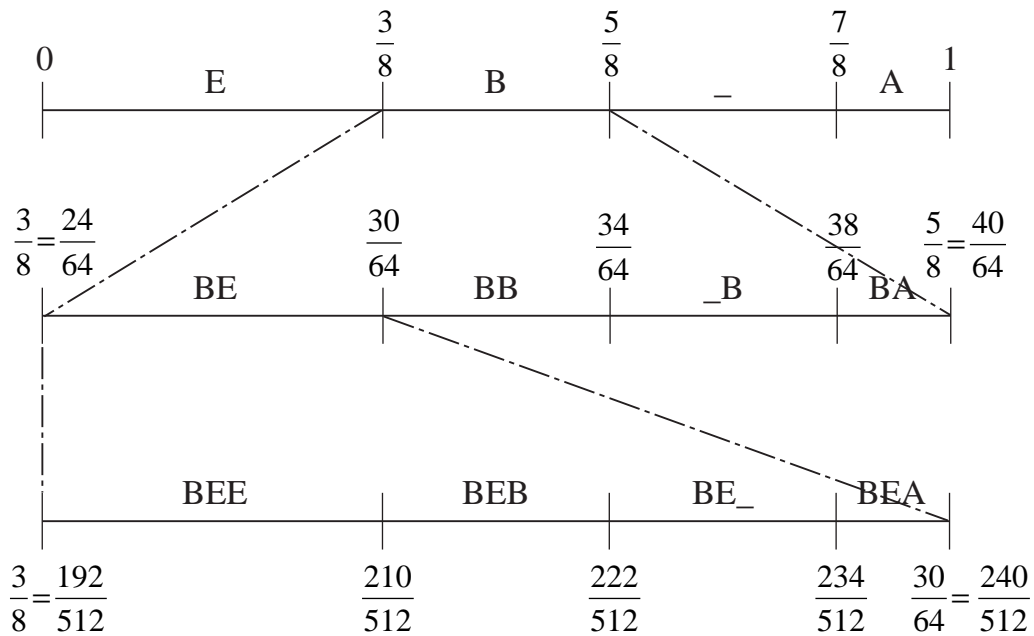


Our string starts with a 'B', so we take the 'B' interval and divide it up again in the same way:



The boundary between 'BE' and 'BB' is $\frac{3}{8}$ of the way along the interval, which is itself $\frac{2}{8}$ long and starts at $\frac{3}{8}$. So the boundary is $\frac{3}{8} + \left(\frac{2}{8}\right) \times \left(\frac{3}{8}\right) = \frac{30}{64}$. Similarly the boundary between 'BB' and 'B_' is $\frac{3}{8} + \left(\frac{2}{8}\right) \times \left(\frac{5}{8}\right) = \frac{34}{64}$, and so on.

The second letter in the message is 'E', so now we subdivide the 'E' interval in the same way. We carry on through the message ...



and, continuing in this way, we eventually obtain:



So we represent the message as any number in the interval

$$\left[\frac{7653888}{16777216}, \frac{7654320}{16777216} \right)$$

However, we cannot send numbers like $\frac{7654320}{16777216}$ easily using a computer. Computers use binary numbers – a system where all the numbers are made up of 0s and 1s. The first few whole numbers in binary are 1, 10, 11, 100, 101, ... but how do they actually work?

In decimal notation, the rightmost digit to the left of the decimal point indicates the number of units; the one to its left gives the number of tens; the next one along gives the number of hundreds, and so on.

So

$$7653888 = (7 \times 10^6) + (6 \times 10^5) + (5 \times 10^4) + (3 \times 10^3) + (8 \times 10^2) + (8 \times 10) + 8$$

Binary numbers are almost exactly the same, only we deal with powers of 2 instead of powers of 10. The rightmost digit of a binary number is units (as before); the one to its left gives the number of 2s; the next one the number of 4s, and so on.

$$\begin{aligned}
 \text{So } 110100111 &= (1 \times 2^8) + (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) \\
 &\quad + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2) + 1 \\
 &= 256 + 128 + 32 + 4 + 2 + 1 \\
 &= 423 \text{ in denary (i.e. base 10)}
 \end{aligned}$$



Exercise 1

- a) *Write the following denary numbers in binary.*
- i) 56 ii) 147 iii) 623
- b) *What are these binary numbers in base 10 ?*
- i) 11011 ii) 1000011 iii) 101010101

We can write fractions in binary too. In denary, i.e. base 10, the first digit after the decimal point gives the number of tenths; the next gives the number of hundredths, etc. In binary, the first digit after the floating point gives the number of halves, the next the number of quarters, etc. For example,

<i>Fraction</i>	<i>Binary</i>	<i>Fraction</i>	<i>Binary</i>
$\frac{1}{2}$	0.1	$\frac{3}{4} \left(= \frac{1}{2} + \frac{1}{4} \right)$	0.11
$\frac{1}{4}$	0.01	$\frac{3}{8} \left(= \frac{1}{4} + \frac{1}{8} \right)$	0.011
$\frac{1}{8}$	0.001	$\frac{5}{8} \left(= \frac{1}{2} + \frac{1}{8} \right)$	0.101



Exercise 2

- a) *Write the following fractions in binary.*
- i) $\frac{1}{16}$ ii) $\frac{5}{16}$ iii) $\frac{9}{24}$
- b) *What are these binary numbers as decimals in base 10 ?*
- i) 0.1101 ii) 0.10101 iii) 0.100011

But how do we write the number $\frac{7653888}{16777216}$ in binary?

Think about how we would write it in denary (decimal). A good start is to write the fraction in its simplest form. In this case this is easy to do because the denominator is a power of 2 (in fact, $16777216 = 2^{24}$). So to simplify the fraction we just divide the numerator by 2 as many times as we can until we get an odd number, then divide the denominator by the same amount.

So

$$\frac{7653888}{16777216} = \frac{2^9 \times 14949}{2^{24}} = \frac{14949}{2^{15}} = \frac{14949}{32768}$$

If we were calculating this in decimal, we would do it by long division. However, in binary, it is actually a good deal simpler because the denominator is a power of 2.

To understand why, think about how you would write $\frac{14949}{100000} = \frac{14949}{10^5}$ in decimal: it's

just 0.14949. And $\frac{14949}{10^6}$ is simply 0.014949. In general, $\frac{14949}{10^n}$ is 14 949.0 with the decimal point moved to the left by n places.

We can use exactly the same method in binary – all we have to do is convert 14 949 to a binary number and then, as the denominator is $32768 = 2^{15}$, we move the floating point to the left by 15 places.

But how do we convert a number to binary? Start by finding the largest power of 2 which is less than or equal to our number, then subtract it, keeping a note of which power of 2 it was. Then repeat with the remainder, and so on. So ...

n	2^n	Test	Include n ?	Calculate remainder
13	8192	14 949 > 8192	Yes	14949 – 8192 = 6757
12	4096	6757 > 4096	Yes	6757 – 4096 = 2661
11	2048	2661 > 2048	Yes	2661 – 2048 = 163
10	1024	613 < 1024	No	
9	512	613 > 512	Yes	613 – 512 = 101
8	256	101 < 256	No	
7	128	101 < 128	No	
6	64	101 > 64	Yes	101 – 64 = 37
5	32	37 > 32	Yes	37 – 32 = 5
4	16	5 < 16	No	
3	8	5 < 8	No	
2	4	5 > 4	Yes	5 – 4 = 1
1	1	1 = 1	Yes	

So $14949 = 2^{13} + 2^{12} + 2^{11} + 2^9 + 2^6 + 2^5 + 2^2 + 1 = 11101001100101$ in binary (corresponding to the pattern of 'Yes's' above).

Therefore

$$\frac{7653888}{16777216} = \frac{14949}{32768} = 0.011101001100101 \text{ in binary.}$$

However, we can represent the message BE_A_BEE as *any* number in the interval

$$\left[\frac{7653888}{16777216}, \frac{7654320}{16777216} \right)$$



Exercise 3

Show that the binary representation of $\frac{7654320}{16777216}$ is 0.01110100110010111011.

How do we choose a number to represent our message? To get the best compression, we want to send the least possible number of binary digits so we want the shortest number between

0.011101001100101

and

0.01110100110010111011

In this case, the first point in the interval has the shortest binary representation out of all the points in the interval, so we choose this to represent our message. (Note that it is not always an endpoint – in general it is the number in the interval whose numerator can be divided by 2 the most times.)

So BE_A_BEE \equiv 011101001100101

This is 15 digits long as there is no need to send the first zero or the floating point.

Is this an effective compression? If we were using ASCII computer codes, the message would be $8 \times 5 = 40$ digits long, so there is a significant improvement using the Arithmetic Coding method.



Activity 1

Design a Huffman code for this message. How many digits would be in the message?



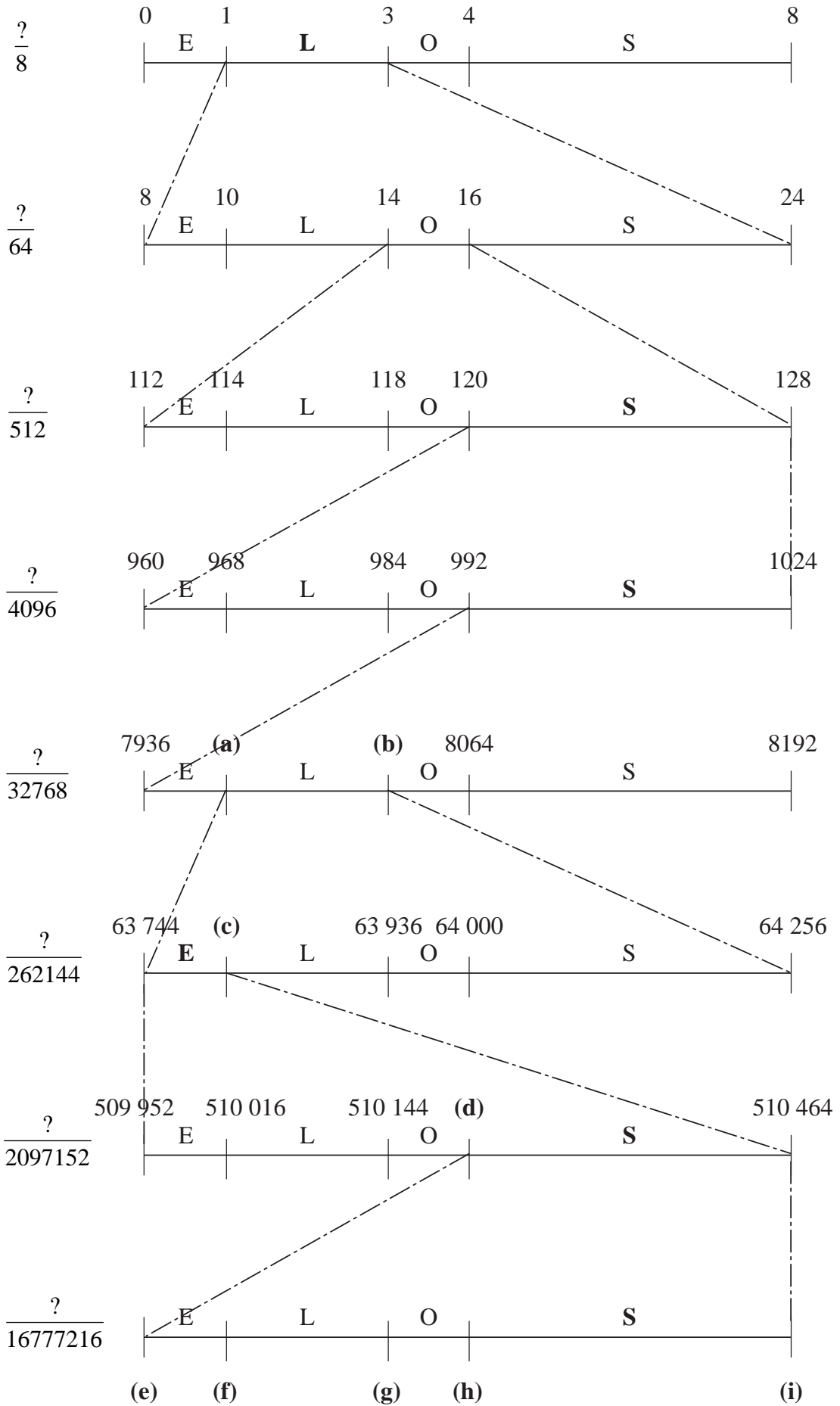
Exercise 4

We want to compress the word LOSSLESS using arithmetic coding.

The frequency counts for the characters in this message are:

E	L	O	S
1	2	1	4

(a) – (i) Fill in the gaps indicated by the letters in brackets on the following model. (For ease of presentation, only the numerators of the fractions are written on the diagram – the denominator for each row is at the side.)



- j) *What interval represents the word LOSSLESS using this model?*
- k) *What is the interval in binary?*
- l) *What is the shortest binary representation of LOSSLESS that we can send using this method?*



Exercise 5

You receive the following message which has been compressed using arithmetic coding:

0110100110101

You know that the following characters were sent:

3 As, 1 C, 1 D 1 I and 2 Ns.

Decode the message.

- a) *What fraction does the number 0110100110101 represent? [Hint: to make the following calculations easier, scale this fraction up so that the denominator is 16777216.]*
- b) *Divide up the interval $[0, 1)$ according to the character frequencies, as in the example. Find where on this line your answer to a) is, and hence find the first letter of the message.*
- c) *Subdivide the interval with your answer to a) in it. Where in this new interval is your answer to a)? Hence find the second letter of the message.*
- d) *Continue subdividing the intervals in this way until you have found all 8 letters. What is the message?*