Reducing File Sizes

As described in the Tao of Computing, computers compress the size of files to save space on the disk and to make them faster to send to someone else on a network. There are two types of compression, lossless and lossy. In lossless no information is lost whereas in lossy you can lose information. PostScript ins lossless whereas GIF and JPEG is lossy.

An important question is how do you decide the codes to represent a file so it is as small as possible. Let's consider lossless compression for now and only do text. Take the line

to be or not to be

If you uses standard Unicode that uses 16 bits for each letter, it takes 16 * 18 = 288 bits or 36 bytes (2 * 18). Now for this expression you can do much better. There are only 7 unique letters: b e n o r t. You can use 3 bits since $2^3 = 8$ so you could store 8 unique values. This is 1 more than you need for the 7 you have. You could use the codes

b	e	n	o	r	t	space
000	001	010	011	100	101	110

This would give a file size of 3 * 18 = 54 bits. This is much smaller than 288.

Can you do better? Yes you can. The above examples used fixed size codes of either 8 or 3 bits. You can make some code smaller and some larger. Try to see how small you can make the file by choosing the codes and size carefully for the letters given. Try this in groups to see what you can come up with.

Group Alice Programming

In groups of 2-4, perform exercise 3 on p. 111 (Gallop and Jump) from the Learning to program in Alice textbook.

Computers & Numbers

As described in the Tao of Computing, computers use binary or base 2 to represent numbers. Computer scientists also use octal or base 8 and hexadecimal or base 16. They are all powers-of-two because computer hardware stores values as 0 or 1 (binary). This exercise is designed to help understand the different representations and how they are related.

Recall each digit in a number represents the base raised to the power of its position from the right in the number where you start with 0. Thus, 485 in base 10 is

$4 * 10^2 + 8 * 10^1 + 5 * 10^0$

400 + 80 + 5

485

Base 2 or binary is the same idea of powers-of-2. Also, you can only have 0 or 1 as values in a number. Thus, 1011 in base 2 becomes

$1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$

1 * 8 + 0 * 4 + 1 * 2 + 1 * 1

8 + 0 + 2 + 1

11 in base 10

This is quite different than thinking of 1011 as as base 10 number which is one thousand and eleven.

Octal or base 8 uses the same ideas. You can only have values from 0 to 7 and you use powers-of-8. Thus, 517 in base 8 becomes

$5 * 8^2 + 1 * 8^1 + 7 * 8^0$

5 * 64 + 1 * 8 + 7 * 1

320 + 8 + 7

335 in base 10

Hexadecimal or base 16 is the same. There is a small issue of how to represent the 16 values from 0 to 15. You use the same symbols for 0 through 9. For 10 you use A, for 11 you use B, 12 is C, 13 is D, 14 is E, and 15 is F. Thus, F6B in base 16 becomes

$15 * 16^2 + 6 * 16^1 + 11 * 16^0$

15 * 256 + 6 * 16 + 11 * 1

3840 * 96 + 11

3947 in base 10

Now you will try to convert your own numbers to different bases. You need 8 people (or the leftmost digits without people become 0). Line up in a row. Keep in mind that right and left are as you look at the number so the person on your right is a higher power – backward from the above discussion. This makes it a little tricky. The rightmost person (farthest left from line's perspective) will be the first binary digit and the leftmost person (farthest right from line's perspective) will be the last binary digit. Your peer leader will give you a number in binary and each person should figure out what value (0 or 1) that they represent. Write it on a piece of paper and hold it up. Have everyone check that the number down the line is correct. Go ahead and do this now.

The peer leader will give you the future directions since you will be in line.

The next step is to convert the binary number to decimal. Start with the person on the right who will be zero. The next person to the left adds one to the value from the previous person. Each person says their number and remembers it. Now take your number, raise 2 to that power and multiply it by the digit you have written you your piece of paper. Write it on your piece of paper.

When everyone is done, start with the leftmost person, and say your number. The next person to adds this value to their own value (in base 10) and says the value. Continue down the line until you get to the rightmost person. Check to see if you have the correct number and fix as necessary.

Now we will convert this value to hexadecimal or base 16. Each 4 binary digits represent 1 hexadecimal digit (since 2^4 is 16). The leftmost and rightmost groups of 4 should work together to figure out their hexadecimal value. You can convert your 4 digits to decimal and then get the hexadecimal value.

Now convert your values to base 10 to check. The rightmost group is 16^0 digit times their value and the leftmost group is 16^1 times their value. If this goes correctly you will get the same value as before in decimal.

The last step is to convert to octal or base 8. Create groups of 3 where the leftmost group will only have two people. Count the missing person as being a zero in the leftmost digit for their group. Now convert each group to octal. Then convert to decimal.