Gray and anti-Gray codes

Besides counting, there are other algorithms for iterating through a sequence of n-digit representations. One of those algorithms that is useful in computing is a Gray code.

A Gray code is any numerical code where consecutive integers are represented by binary numbers that differ in exactly one digit. If we look at our binary counting example, we can quickly see that the standard binary representation is not a Gray code. Specifically, consider three (011) and four (100). They are consecutive integers, but they differ in *every* digit, whereas consecutive elements in Gray code may differ in exactly one digit.

Constructing a Gray code

There are several Gray codes, but we will look at one in particular: the *binary-reflected Gray code*. We can build an $(n + 1)$-bit Gray code from an n-bit Gray code in a few simple steps.

1. Copy the sequence (creating an 'original' and a 'copy')
2. Reverse the order of the elements in the 'copy' sequence (hence the name binary-*reflected* Gray code)
3. Prefix each element in the 'original' sequence with a '0'
4. Prefix each element in the reversed 'copy' with a '1'
5. Concatenate the 'original' sequence and the 'copy' sequence

If we start with the Gray code for 1-bit, we can iteratively build a Gray code of any size by repeating these steps. Luckily, the $n = 1$ case is easy: 0, 1.

Initial Sequence	**Copy the Sequence**	**Reflect the copy**	**'0' + original '1' + copy**
0	0	0	00
1	1	1	01
	0	1	11
	1	0	10

Initial Sequence	**Copy the Sequence**	**Reflect the copy**	**'0' + original '1' + copy**
00	00	00	000
01	01	01	001
11	11	11	011
10	10	10	010
	00	10	110
	01	11	111
	11	01	101
	10	00	100

Initial Sequence	**Copy the Sequence**	**Reflect the copy**	**’0’ + original ’1’ + copy**
000	000	000	0000
001	001	001	0001
011	011	011	0011
010	010	110	0010
110	110	110	0110
111	111	111	0111
101	101	101	0101
100	100	100	0100
	000	100	1100
	001	101	1101
	011	111	1111
	010	110	1110
	110	010	1010
	111	011	1011
	101	001	1001
	100	000	1000