Gray code

Table of Contents

Gray code is a binary numeral system where two successive values differ in only one bit.

For example, the sequence of Gray codes for 3-bit numbers is: 000, 001, 011, 010, 110, 111, 101, 100, so $G(4) = 6$.

This code was invented by Frank Gray in 1953.

Finding Gray code

Let's look at the bits of number n and the bits of number $G(n)$. Notice that i-th bit of $G(n)$ equals 1 only when i-th bit of n equals 1 and $i + 1$-th bit equals 0 or the other way around (i-th bit equals 0 and $i + 1$-th bit equals 1). Thus, $G(n) = n \oplus (n >> 1)$:

```
int g (int n) {
    return n ^ (n >> 1);
}
```

Finding inverse Gray code

Given Gray code g, restore the original number n.

We will move from the most significant bits to the least significant ones (the least significant bit has index 1 and the most significant bit has index k). The relation between the bits n_i of number n and the bits g_i of number g:

$$\begin{aligned} n_k &= g_k, \\ n_{k-1} &= g_{k-1} \oplus n_k = g_k \oplus g_{k-1}, \\ n_{k-2} &= g_{k-2} \oplus n_{k-1} = g_k \oplus g_{k-1} \oplus g_{k-2}, \\ n_{k-3} &= g_{k-3} \oplus n_{k-2} = g_k \oplus g_{k-1} \oplus g_{k-2} \oplus g_{k-3}, \\ &\vdots \end{aligned}$$

The easiest way to write it in code is:

```
int rev_g (int g) {
  int n = 0;
  for (; g; g >>= 1)
    n ^= g;
  return n;
}
```

Practical applications

Gray codes have some useful applications, sometimes quite unexpected:

- Gray code of n bits forms a Hamiltonian cycle on a hypercube, where each bit corresponds to one dimension.

- Gray codes are used to minimize the errors in digital-to-analog signals conversion (for example, in sensors).

- Gray code can be used to solve the Towers of Hanoi problem. Let n denote number of disks. Start with Gray code of length n which consists of all zeroes ($G(0)$) and move between consecutive Gray codes (from $G(i)$ to $G(i+1)$). Let i-th bit of current Gray code represent n-th disk (the least significant bit corresponds to the smallest disk and the most significant bit to the biggest disk). Since exactly one bit changes on each step, we can treat changing i-th bit as moving i-th disk. Notice that there is exactly one move option for each disk (except the smallest one) on each step (except start and finish positions). There are always two move options for the smallest disk but there is a strategy which will always lead to answer: if n is odd then sequence of the smallest disk moves looks like $f \to t \to r \to f \to t \to r \to \ldots$ where f is the initial rod, t is the terminal rod and r is the remaining rod), and if n is even: $f \to r \to t \to f \to r \to t \to \ldots$.

- Gray codes are also used in genetic algorithms theory.

Practice Problems

- SGU #249 **"Matrix"** [Difficulty: medium]