Notes on Gray codes

Although determining if a graph contains a Hamiltonian cycle is a NP-complete problem, the problem may still become easy (solvable in polynomial time) if we restrict the input to certain classes of graphs.

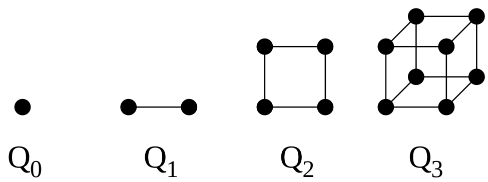
For example, if the input graph G is restricted to only have vertices of degree at least $|V(G)|/2$ and to have at least 3 vertices then by Dirac's theorem, the algorithm can just always output "yes".

Here is another class of graphs which always have a Hamiltonian cycle.

Definition 1. The *hypercube graph* Q_n is defined recursively as follows.

$Q_0 = K_1$, the only graph with one vertex.

Q_n is obtained from taking the disjoint union of two copies of Q_{n-1} and joining vertices which are (labelled) the same in both copies of Q_{n-1}.



Remark 1. *Q_n has 2^n vertices.*

All vertices of Q_n have degree n.

Vertices of Q_n can each be labelled with a different n-bit binary string so that two vertices have an edge between them if and only if their labels differs in exactly one bit.

Theorem 1. *For $n \geq 2$, Q_n has a Hamiltonian cycle.*

Proof. Q_n has a Hamiltonian cycle if and only if there is a way to label all n-bit binary strings into a sequence such that consecutive elements of the sequence differ in exactly one bit and the first and last element differ in exactly one bit.

Such a sequence exists and is called a *Gray code.*

The Gray code for 2-bit binary strings is

$$00, 01, 11, 10$$

.

The n-bit Gray code can be obtained from the $n-1$-bit Gray code by concatenating

- the $n-1$-bit sequence forward with 0 added as the first bit, and
- the $n-1$-bit sequence backwards with 1 added as the first bit

. Thus, the 3-bit Gray code is

$$000, 001, 011, 010, 110, 111, 101, 100$$

Now we need to check that Gray codes indeed have the properties we need. That is, consecutive strings differ in exactly one bit and the first and last string differ in exactly one bit.

We can simply check this property for 2-bit binary strings (00 and 01 indeed differ in one bit, 01 and 11 indeed differ in one bit, etc).

Now given that the $n-1$-bit Gray code has the properties we want, we show that the n-bit Gray code also has this property. Consecutive elements in the first half of the sequence indeed have this property since adding a 0 as the first bit does not change the number of bits in which they differ. Similarly, consecutive strings in the second half of the sequence differ in exactly one bit (the property of differing in one bit is symmetric so having the sequence backwards does not change this). Now the last element of the first half and the first element of the second half of the sequence differ in exactly one bit because they are the same $n-1$-bit string (namely, the last string of the $n-1$-bit Gray code) where one has a 0 added and the other has a 1 added. The same is true for the last element of the sequence and the first element of the sequence (they are both the first element of the $n-1$-bit Gray code but one has a 1 added and the other has a 0 added). $\square$

Gray codes have many applications from error correction (especially when variables can be modified while they are being read) to solving the Tower of Hanoi puzzle (the bit which changes indicates the size of the disk to move).