QUANTUM NETWORKS

We begin by considering a class of random Boolean networks, invented and popularized by theoretical biologist Stuart Kauffman, called NK networks. Specifically we consider a graph with N nodes, which are populated by Boolean variables. To each of these nodes we assign $1 \leq K < N$ other nodes, with equal probability across the $\binom{N-1}{K}$ possibilities.

The K nodes connected to a particular other node serve to determine that node's future evolution. In particular, each node is also associated with a randomly chosen Boolean expression with K inputs and 1 output. The values (0 or 1) of the K inputs to the node in question are fed into the corresponding Boolean expression, generating a value (0 or 1), which then becomes the value of the node for the next time unit. There are 2^{N2^K} assignments of Boolean expressions to the nodes, and one of these is chosen with equal probability.

Once this upfront setup is accomplished, we have a particular instance of an NK network. Its evolution proceeds deterministically from its initial conditions (i.e. the initial assignment of 0s and 1s to the nodes). Over time, the nodes flash ON (1) or OFF (0) in complicated patterns. The resulting dynamical system can be in any of a^N states at every point in time. Since the state space is finite, we know that eventually the system will return to a state it visited before, and after that it will be caught in a periodic cycle. However, the number of states is usually astronomically large...

In cases where the return periods are very long, the system would appear chaotic and no pattern would be discernible. Thus, for systems of this type, it is very important to distinguish between long and short attracting cycles. Perhaps quantum computing can help to tame this exponential complexity. Consider a game where your opponent gives you an instance of an NK network and you need to put forth a quantum algorithm to simulate it.

1) How many qubits do we need to represent an NK network with $N = 5$ and $K = 1$?

2) Can you design a *universal* quantum simulator, that can work when $N = 5$ and $K = 1$, irrespective of the specific instance of the NK network?

3) How does your answer to question 1 above change if K is increased to 2?

4) How does the number of qubits needed to universally simulate NK networks increase with N and K?