CS 7A - Spring 2016 - Elementary Cellular Automata I. Due 4/28/16

It is theorized that the underlying structure of the universe is not actually a continuum–that this is an illusion produced by a very fine multi-dimensional mesh of cellular automata. An inter-linking net of discrete cells whose behavior is determined by a dynamic involving each cell's state and the states of its neighbors.

Here we investigate the simplest model of cellular automata studied by Steven Wolfram In the 1980s: one-dimensional cellular automata, or what he calls elementary cellular automata. Wolfram published *A New Kind of Science* in 2002, claiming that cellular automata have applications in many fields of science, including computer processors and cryptography.

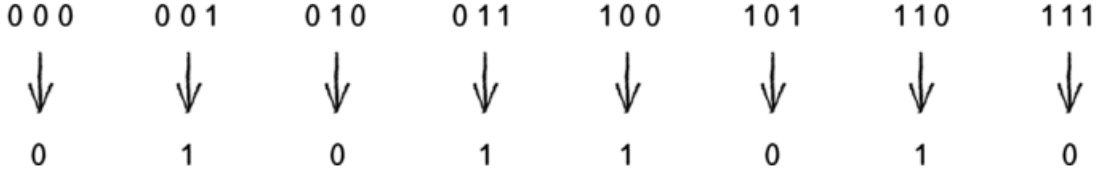
Elementary cellular automata consist of a 1-dimensional **grid** of cells, each cell in one of two **states** (say, 0 or 1) and having a **neighborhood** (left, center and right). There is an initial state of the first generation, which might look like this:

1	0	1	0	1	1	1	0	0	1

A cell's state in generation $n+1$ depends on the cells neighborhood in generation n:

$$C[n+1] = f(\text{C[n]'s neighborhood})$$

Now there are 8 possible neighborhoods and so there are 2^8 different rules that can apply. This one for instance:



The rule assigns to each of the possible neighborhoods, one of the two possible states.

1. Write code involving a class called `CA` to hold a vector of bool for the cells, a vector of bools for the rule set (use the rule set shown above) and a constructor to initialize the first generation to have 79 cells, all zeros except for a 1 in the middle.

2. Write a function called `generate` to compute the next generation, and a driver to implement `CA` and show how the sequence of generations evolves.