PHY371 ProjectVI
Exercises on Cellular Automata, Recursion and Class

Instructor: Ying Tang
Department of Physics
Gordon College

November 14, 2014

1 Introduction

In project VI, you have three exercises. You should write a seperate python program for each exercise. The purpose of these exercises is to help you understand more about Class, Cellular Automata simulations, and gain some first-hand experience in designing and implementing recursion.

You are encouraged to discuss these exercises with others. However, your final programs should be original and reflect your own work. Your grade will be based mainly on the correctness and the structure of your programs.

2 Project 1

(The same as in-class exercise of lecture 18) Write a cellular automata (CA) program with Rule 90 (see Fig.1) to produce Pascal's triangle. Print out the triangle up to 50 rows.

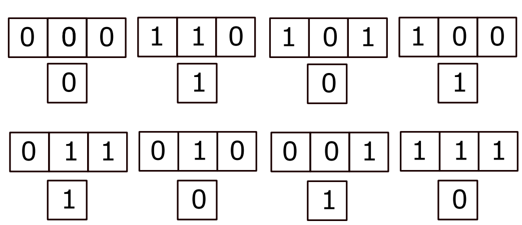

Figure 1: Rule 90 for 1D Cellular Automata simulation.

3 Project 2

Pascal's triangle is shown in Fig. 2. Find out hidden rules behind this triangle, then write a program to reproduce this triangle using a **recursive function**. Your program should print out this triangle to its 10th row.

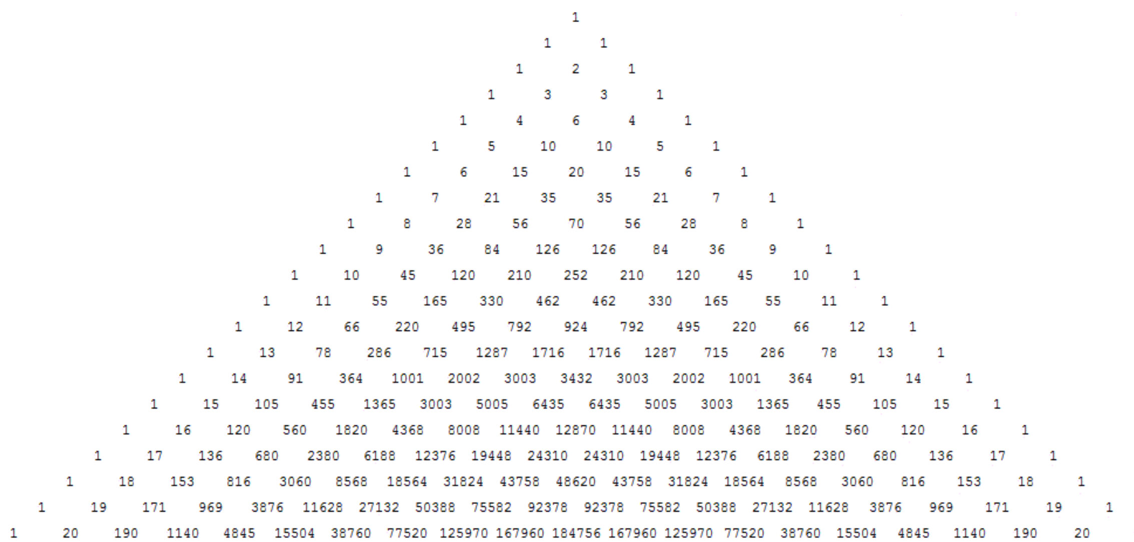

Figure 2: Pascal's triangle. You might have noticed that, if you color all the odd numbers in this figure, you will obtain the same triangle of Project 1 (which has an official name — Sierpinski triangle)

4 Project 3

Use **class** to simulate John Conway's *The Game of Life*.

Your working grid has a size of $n_x \times n_y$. n_x and n_y should be given from the terminal. Each element of your grid is a cell. There are only two possible states of your cell: alive or dead. You start from certain initial configuration of cells and let cells evolve with time. For each time step, cells interact with its eight nearest neighbors (Moore Neighborhood). The interation rules are as follows:

1. Any live cell stays alive if it has either 2 or 3 live neighbors.
2. Any live cell dies if it has less than 2 or more than 3 live neighbors.
3. Any dead cell with precisely 3 live neighbors springs to life.

Program tasks:

1. Your simulation should include at least one glider (given in the sample code already), a gun and a few oscillators. These patterns have been discussed in the lecture note. You can also find these patterns online (`http://en.wikipedia.org/wiki/Conway's_Game_of_Life`).

2. Your program should show an animation of your simulation results.

   A sample program "gameofLife.py" has been uploaded to our Github course repository. Feel free to use it as you wish. You are also encouraged to write your own program.

5 Submission

When you submit your programs, please make sure your programs generate all expected results as described above. Make sure your programs are ready to go without any further modifications. If you think I should be aware of a specific input or features of your program, please write a seperate note to explain and send it to me together with your programs.

Please submit three python programs to ying.tang at gordon.edu by **6:00pm on Friday 11/21**. A late submission will cause a 30% deduction of your grade.

If you have any question about this set of exercises, please don't hesitate to let me know.