Lights Out

1 Problem description

The goal of this project is to use linear algebra concepts to describe the game "Lights Out." The game is played as follows. You have a 5×5 grid of lights, some of which are turned on and some of which are turned off. By pushing one button you change the state of that button and its four neighbors (above, below, left, and right). Note that the lights in the top row, bottom row, left column, and right column have fewer than four neighbors. See Figure 1 below for a sequence of moves. The goal of the game is to find a sequence of moves which turns off all the lights. Note there are some initial configurations such that it is not possible to turn off all the lights.

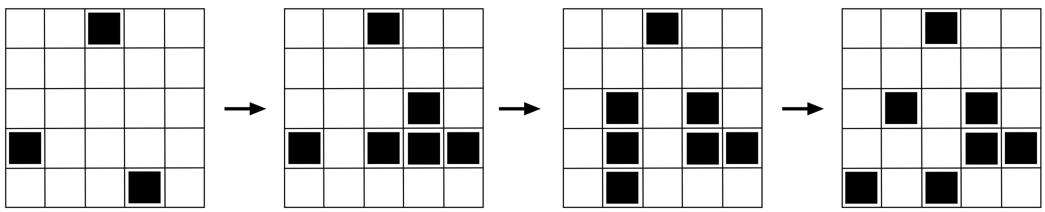

Figure 1: *A sequence of Lights Out board configurations. The initial state is shown on the left: three lights are off while the other 22 are on (a black square represents an off-light and a white square represents an on-light.). The subsequent board configurations are produced by pushing the buttons in the following order: 4th row/ 4th column, followed by 4th row/ 2nd column, followed by 5th row/ 2nd column.*

2 Questions to consider

Your task is to describe this game mathematically using concepts from linear algebra. Work through the following questions to help you formulate a complete description of Lights Out.

(i) Describe a Lights Out board in terms of vector spaces and fields.

(ii) What is the linear transformation? Hint 1: the transformation should be a linear map between the space of "move sequences" and the space of "light configurations." Hint 2: Use the word "adjacency matrix" (not covered in class, but covered in your book at the end of Section 2.3, except that they call it an "incidence matrix")

(iii) There are some initial configurations i such that it is not possible to turn off all the lights. In other words, there are some i such that

$$i + Tm \neq f$$

for any sequence of moves m (where f is the vector of all "off"'s). Formulate the question "For which i is it not possible to turn off all the lights" as a question about the range and nullspace of the linear transformation T (Hint: if $i + Tm = f$ then $Tm = f - i$. Furthermore, $f = 0$ and in Z_2^n, the vector $-x = x$.).

(iv) Write a program in Python that determines whether an initial board configuration can be solved.

3 Discussion section structure

Week 1:
(a) Discuss the game "Lights Out"

(b) Briefly discuss principles of debugging programs

(c) Write a Python function to "Find My Neighbors" for a general $m \times n$ matrix using **if/ then/ else statements**

Week 2:
(a) Discuss adjacency matrices

(b) Learn to create matrices, and store and access the elements of a matrix, using numpy

(c) Write a Python function to build the adjacency matrix for Lights Out using **for loops**

Week 3:
(a) Learn some matrix operations: switching rows, adding rows

(b) Write a Python function to put any $m \times n$ matrix over the field Z_2 in reduced row echelon form using **while loops**. The following test cases are useful:

$$A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad A_5 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

(c) In the above, make sure to also keep track of the columns that do and don't have a pivot (i.e., the columns with a pivot are the columns where there is a row in which it is the leading entry); this will be necessary to finding the null space.

Week 4: Note that although all discussion meetings are optional, week 4 is super-extra optional. This week we don't learn any new programming techniques, we just put together all of the elements from weeks 1-3 to solve the game Lights Out.

(a) Write a Python function to compute the null space of any $m \times n$ matrix over the field Z_2, using the reduced row echelon form and the location of the non-pivots.

(b) Given an initial configuration, you can create the augmented matrix and use your RREF function to solve for the sequence of steps required to solve the puzzle. If you end up with an inconsistent matrix, then you know that the system is not solvable.

(c) As part of the above, we will learn how to create an augmented matrix using numpy and how to convert $m \times 1$ matrices into m-dimensional vectors.

(d) Find the null space for the augmented matrix. This is necessary because if the null space is non-trivial then there are multiple solutions.