Project I: Lights Out

Lights out is a one-player game that consists of a grid of squares that can be either "on" (colored red) or "off" (colored yellow). Each time a square is clicked, that square and the four squares immediately north, south, east, and west of it are toggled. You win the game by turning all of the squares yellow. Try it out at

http://www.cs.middlebury.edu/~schar/java/lightsout.html

or

http://www.jaapsch.net/puzzles/javascript/lightjcl.htm

You will need to have the latest version of Java to get the first website to work. If you're having trouble, use the computer lab in CMC 301. The "classic" game is played on a 5x5 board and beginning with all reds. Try to beat this game, just making up your strategy as you go along. Now try with the 2x2 board, and then the 3x3.

If you can't get the first website to work, then try the second one. It only features the 5x5 game, but will sill give you an idea of what's going on.

This project consists of using Linear Algebra to analyze how to win this game, and to answer many other questions one might have about it, such as how many ways there are to win, and what happens if we start with initial configurations other than all lights on.

Let's begin with the 2x2 case, to give us an idea of what's going on. We associate every configuration of the board to a vector with four entries corresponding to the four squares in the board (you may choose any ordering of the squares you want). An entry is 1 if the corresponding light is on (red) and 0 if the light is off (yellow). So when all the lights are turned on, the vector is (1,1,1,1). You win the game by turning all the lights off, that is, arriving at the state (0,0,0,0).

Let's call a *strategy* a sequence of squares that you click on (with each click toggling some of the adjacent squares, of course). Note that any square should be clicked on either once or not at all – clicking twice has no effect – and the order in which you click them makes no difference to the final configuration on the board. We can also represent a strategy by a vector with four entries: an entry is 1 if the strategy involves clicking on that square and 0 if it does not. So the strategy (0,0,0,0) involves doing nothing, while the strategy (1,0,1,1) involves clicking on all the squares except the upper-right-hand one.

Now suppose we begin with all lights on, i.e. at the starting point $\mathbf{b} = (1,1,1,1)$. Suppose we apply a strategy (x_1, x_2, x_3, x_4) to $\mathbf{b}$. Let's isolate the effects on the upper-left corner, which is the first entry in the vector. If $x_1 = 1$, then it toggles the first entry of $\mathbf{b}$. To algebraically represent the toggling, we can add one and then take the result modulo 2 (i.e. 0

if the result is even and 1 if the result is odd). If x_2 = 1, then the first entry of **b** is also toggled, and the same if x_3 = 1. Clicking on the lower-right square does not change the upper-right square, so we may disregard x_4. The result of applying our strategy is thus

$$x\_1 + x\_2 + x\_3 + 1 \pmod 2,$$

where the 1 on the end is the initial state of the upper-left corner, namely the first entry of the vector **b**. So in order to have the end state of the upper-left corner be off (yellow), we need

$$x\_1 + x\_2 + x\_3 + 1 = 0 \pmod 2.$$

Problem 1: Perform similar analyses for effects of applying the strategy **x** = (x_1, x_2, x_3, x_4) to the other squares on the board, beginning with the initial state **b**. Write the result of your analysis in the form of a matrix equation A**x** = **b**. The matrix A is called the *adjacency* matrix.

Problem 2: Determine if there is a solution to the 2x2 puzzle beginning with **b** = (1,1,1,1). If you find there is one, try it out on the web applet and make sure it works.

Problem 3: Is there a solution to the 2x2 puzzle no matter what the initial state? Give a solution (in vector form) for the initial state with the upper left and lower right squares on and the others off.

Let's now move up to a slightly larger game: a 2 x 3 lights out game: two rows and three columns. (Unfortunately, the web applet above doesn't have this size available.)

Problem 4: Determine whether there is a solution to the 2x3 puzzle beginning with all the lights on. Is this solution unique? Usually having free variables means that we get an infinite number of solutions; however, in this particular problem our scalars are restricted to just the values of 0 and 1. With this in mind, how many different solutions are there to the puzzle? Write them down.

Problem 5: Let A be the adjacency matrix for the 2x3 game. Describe all the solutions to the system **Ax = 0**. What happens to the board when we push all of the buttons described by one of these solutions?

Problem 6: Find the dimensions of Ker(T) and Im(T), where T is the linear transformation whose matrix is **A**. While you're at it, go ahead and find bases for Ker(T) and Im(T). If possible, find a vector **b** that is not in Im(T). What does such a vector **b** represent in the Lights Out puzzle?

Problem 7: True or False: In a general Lights Out puzzle, if a solution to the puzzle from a given starting state is unique, then it is possible to solve every puzzle now matter how the lights are initially configured. Explain in some detail.

Now you're ready to move on to larger boards, namely the 3x3, 4x4, and 5x5 boards. The questions you should consider in each case:

i) What is the rank of the adjacency matrix?

ii) Can we solve the puzzle if we start with all of the lights on? If so, write down a solution (i.e., a list of which buttons to push.)

iii) Is the solution unique?

iv) Can we solve *every* puzzle no matter how the pattern of lights is initially configured? If not, how many initial configurations are solvable? How many are not solvable?

Whatever additions you pursue to these types of questions will be welcomed and rewarded. For instance, how do you think the answers to the above four questions will change for an nxn board as n grows? Can you give some data for various board sizes (if you enjoy programming), or prove the existence of patterns for some families of board sizes? Here is another specific problem you might want to consider:

Problem 8 (bonus): Prove that it is always possible to win the game (regardless of the size of the board) if you start with all the lights on.

A note on computation: For the 3x3 case and larger cases, the matrices involved are quite large, and it is necessary to use a computer algebra system to work with them. One option for doing this is to use Mathematica, which is installed in the computers in the CMC (I recommend using room 301, where you'll always be able to find a free machine). Double click on the Mathematica icon on the desktop, and then open a new notebook. Here's an example of how to create a matrix:

```
M = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
```

To execute this command, you need to hold down the shift key and hit return.

Once you execute this command, Mathematica will show you the output, though only as you typed it and not in matrix form. If you want to see it in matrix form, type MatrixForm[M]. Remember to hit shift-return to execute. Now comes the part that computers are so good for:

RowReduce[M, Modulus->2] – returns the reduced row-echelon form of the matrix M, working modulo 2.
NullSpace[M, Modulus->2] – returns a basis for the kernel of M, again working modulo 2.

You can find more matrix-related commands in the help menu.

Instructions for the writeup: you should explain as carefully as you can the steps you've taken to solve each part of the project. Write in complete sentences and give details. You'll be graded 75% on the mathematical accuracy of what you've done and how far you pushed the project – extensions you come up with will be rewarded – and 25% on the clarity of your write-up (25%). Both handwritten and word-processed writeups are fine. You should include the results of your computer calcuations, but you don't need to hand in the code you used.