Etude No.9 Lights On and Off

Chenrong Lu
Suhaib Aljehani
University of Otago

Etude No.9, lights on and off is a question of interest. It is intriguing in the fact that there does not always exist a guaranteed solution by solving a linear system of equation(s). Hence there is the need to develop an algorithm which is capable of checking all cases of valid solutions, while maintaining efficiency in computation time, as well as memory usage. The general problem formulation is such that, given a set of lights : A B C D...
and a set of connections:
AB AC DC ...
(Such that A turns off B,C and as well as it self)
Figure out a configuration [A=On B=Off C=On...] which gives a result: [A=Off,B=Off,C=Off]
Some differences of this problem to pushing buttons.

1) In contrast to pushing buttons, lights are allowed one way connections, as opposed to two way.
2) A light which is off can be turned, as opposed to a button which is down.
3) The order does not matter, and a switching a light on and off and back on again does not change anything.

The following text describes a brief overview of our solution and the implemented algorithm

I. Algorithmic solutions

A. Naive Algorithm

The naive solution would be to try every solution, change the state of the lights, and see if the plan results in a end state with all lights off.
Worst Case: O(2^N)
$BestCase : O(2^N)$

B. Our Algorithm

An optimized way, which is our implementation, realizes that you can represent the connections of the lights as a graph, represented in an adjacency matrix , and a plan which is represented with a diagonal matrix of 1 or 0 entries.The result of the plan is the result of the matrix multiplication of the adjacency matrix, and the bit-wise XOR of the column entries. In the worst case we still check for all cases, but our implementation ended up to be a combination of all the plan, and stops early if a solution is found. This implementation guarantees to find solutions for adjacency matrices.

C. Matrix solution

Alternatively, we could attempt to find solutions for a plan, given a desired result of solution. However, not all adjacency matrices are invertible.

D. Testing and running

To test the code, we wrote a brief python script, generating formatted test cases. Of a specified number of lights and number of test cases.
gen_test.py write
generates the test cases and stores it into a file named ”test.txt”
gen_test.py exec
runs the ./lights cpp program, and feeds in test data line at a time.
Alternatively, run ./lights and input in cases manually as according to the etude specifications.

II. Conclusions and reflections

A. Reflections

Through this etude, we have gained new experience, in formulating a problem to both algorithmic as well as mathematical solutions. We also have seen how a problem can be similar in nature to

another problem, and how it could also be subtlety different.

B. Team work

Team work was divided mainly to : Chen: Writing and testing the algorithm. Suhaib: Writing the algorithm boilerplate, and designing the algorithm, as well as testing.