EGCO 221 – Group Project 2 (Light Out Puzzle)

*** The project can be done in a group of 2-4 students (** group of 1 student is not allowed). But each group must do it by themselves. **Everyone involved in cheating will get ZERO point**

Lights are placed in a N x N grid. Each of them can be ON (1) or OFF (0). When a light is toggled, from ON to OFF or OFF to ON, so are its horizontal and vertical neighbors – if any of them exists.

Find a way to switch off all lights and print one possible sequence.

There are many approaches to solving this puzzle. One solution is to encode N x N grid as a binary number. For 3 x 3 grid, there are 2^9 or 512 states: from 000000000 (=0) to 111111111 (=511). Therefore, the initial state in this example is 000110101 (=53)

Construct a graph with all possible states being nodes. An edge between u and v exists if a toggle of u's bit results in v. The problem can then be reduced to finding a shortest path from the initial state to 000000000.

```
Enter number of rows = 3
Grid size = 3 x 3 = 9
Enter initial states (9 bits) = 000110101

      | col 0 | col 1 | col 2 |
row 0 |   0   |   0   |   0   |
row 1 |   1   |   1   |   0   |
row 2 |   1   |   0   |   1   |

5 moves to turn off all lights

>>> Move 1 : turn on row 1, col 2
      | col 0 | col 1 | col 2 |
row 0 |   0   |   0   |   1   |
row 1 |   1   |   0   |   1   |
row 2 |   1   |   0   |   0   |

>>> Move 2 : turn on row 2, col 1
      | col 0 | col 1 | col 2 |
row 0 |   0   |   0   |   1   |
row 1 |   1   |   1   |   1   |
row 2 |   0   |   1   |   1   |

>>> Move 3 : turn off row 2, col 2
      | col 0 | col 1 | col 2 |
row 0 |   0   |   0   |   1   |
row 1 |   1   |   1   |   0   |
row 2 |   0   |   0   |   0   |

>>> Move 4 : turn on row 0, col 1
      | col 0 | col 1 | col 2 |
row 0 |   1   |   1   |   0   |
row 1 |   1   |   0   |   0   |
row 2 |   0   |   0   |   0   |

>>> Move 5 : turn off row 0, col 0
      | col 0 | col 1 | col 2 |
row 0 |   0   |   0   |   0   |
row 1 |   0   |   0   |   0   |
row 2 |   0   |   0   |   0   |
```

<u>Requirements</u> : the program must be written in Java

1. The progam must at least
   - Ask the user to enter an initial state of N x N lights
     For simplicity, you may limit the maximum value of N
   - Find the minimum number of moves to turn off all the lights
   - Display the light states in each move, until all lights are OFF.

This light-out puzzle is another well-known math problem. You can search & use any algorithm (or even pieces of code), provided that the sources are properly acknowledged. But you'll get more points for using graph approach

2. Write a report in THAI describing at least the following
   - Short user manual
   - The algorithms to switch off all the lights. Use flowcharts, pictures, etc. & give examples to show that you understand how the algorithms work. Source code listing without any explanation isn't counted as a proper report. Explain at least the following:
     - What kind of graph that you use and why, type of node & edge in graph, graph creation
     - Any other data structure that you use and why
     - How you find a sequence of moves to switch off all the lights (it is possible to use other algorithms besides shortest path)
   - Any limitation of your program
   - References. No reference means that you claim you did everything by yourself. If it is found later that you did not, this will be counted as cheating

Marking

2 points All lights are off. Light states are correct in each move

1 point Minimum number of moves

2 points User interface
- This doesn't mean fancy output, but rather how your program is easy to understand & to use. For example, can the user continue playing without having to restart the program ? Are the instructions clear enough ? Is the output properly formatted & readable ? Your program needs not have the same user interface as the example's
- This part must be your own effort. Even if 2 groups take algorithms or pieces of code from the same source, it'd be impossible to have the same/similar coding for the user interface. Therefore, same (or too similar) user interface coding is considered cheating

4 points Programming (2 points) + graph programming with JGraphT (2 points)
- Although this is not a programming course, your program should still be well structured – with proper design of classes, proper use of Java collections, proper exception handling
- Even if you use pieces of code from the Internet. Try converting it to Java & JGraphT version

6 points Report. Points are given based on content and format

Submission : due Tuesday 5 December, 18.00

1. A report in DOC or PDF. The front page must contain names and IDs of everyone in your group
2. Source files (*.java)
3. Put all files in a folder. Name the folder after the ID of one member, zip & submit it to rangsipan@gmail.com
   - Write EGCO 221 – Project 2 as the email subject
   - In case of multiple submissions, only the earliest one will be marked