**Analysis for Problem E - Street Light Pole**

Let x be the number of every possible number of 1s for the rows.

Once x is fixed, y can be determined y = ( x \* number\_of\_rows) / number of columns.

Then assume:

All the rows are vertices on the left side

All the columns are vertices on the right side

Add extra vertex S and edges from S to each row of capacity x and cost 0

Add extra vertex T and edges from each column to T of capacity y and cost 0

Each edge between row i and column j has capacity 1 and:

- if there is a 0 in row i, column j : the cost of the edge is 1

- if there is a 1 in row i, column j : the cost of the edge is -1

After that, min-cost-max-flow algorithm is used to calculate the min number of moves.

- Bellman-Ford-Moore (BFM) shortest path algorithm is used which is O(n^3).

It uses a simple queue in which a node is inserted whenever the optimal distance to it is updated.

- update the flow on multiple vertex disjoint paths after running the shortest path algorithm, this reduces the number of shortest path runs.

- Finally, after running the shortest path algorithm, visit every column and add to a sum the minimum cost of reaching that column times the number of units of flow that column still needs (i.e. the capacity of the edge from the column to T minus the flow on that edge).

If you add this sum to the current minimum cost you will get the minimum possible cost you may still get.

If this minimum possible cost is >= than the minimum cost found so far, then there is no need to continue with the algorithm (it means that the current pair (x,y) which is tested cannot be the optimal one)

*Note:*

*Implementing the solution is really hard even if you understand the procedure.*