# Challenge May 2023

Grid of bulbs

A solution with OPL by Alex Fleischer afleischer@fr.ibm.com

OPL (Optimization Programming Language) is an abstract modeling language that helps model easily optimization problems that can be solved both with IBM CPLEX linear programming and IBM CPLEX constraint programming CPOptimizer (CPO)

```
int n=24;
range r=1..n;

string str="\
000001000000000001110011\
110100010110101000010011\
011101110000001101001110\
000110111000110101101100\
101101011010010011101010\
111000100101110100101000\
110001011100000000000101\
100000010001100000000010\
000110010010110110101001\
011101101011111011100000\
011000101010111011111100\
100011110010000100100111\
000111010010100010001110\
011001010001001111110101\
110001000010111000100000\
000000101100101000101001\
111001010010010011110110\
100000110001111111011010\
110100000011100100110010\
101000110111001110010000\
110000000010011100100101\
111111011011111100010101\
000000000110101011100000\
110001111100000011001111\
";

int ar[r][r];

// turn string s into array ar with scripting
execute parse
{
  var index=0;
  for(var i in r) for(var j in r)
  {
    var p=str.charAt(index);
    index++;
    ar[i][j]=parseInt(p);
  }
}

// first let us forget about
```

```
// the  constraint that
// a lightbulb has to be off to be chosen.

dvar boolean x[r][r] ; // Do we switch this lightbulb ?
dvar int v[r][r]; // Integer decision variable to check that the light if on
finally

subject to
{
  forall(i,j in r)
    (ar[i][j]+x[i][j]+sum(k in r:k!=i)x[k][j]+sum(k in
r:k!=j)x[i][k])==v[i][j]*2+1;
}

tuple t
{
  int i;
  int j;
}

{t} xyok={<i,j> | i,j in r:x[i][j]==1};

// Now let's use some scripting to handle the  constraint that
// a lightbulb has to be off to be chosen.
// For that let's find another one if necessary to turn on and then off

execute
{
  function apply(x,y)
  {
    // The cell has to be off
    if (1==ar[x][y]) fail();
    write("(",y,",",n+1-x,")",",");
    ar[x][y]=1-ar[x][y];
    for(var i in r)
      {
        if (i!=x) ar[i][y]=1-ar[i][y];
        if (i!=y) ar[x][i]=1-ar[x][i];
      }
  }

  function lookforoption(x,y,option)
  {
    for(var k in r) if (k!=y)
    {
      if (ar[x][k]==0)
      {
        option[1]=x;
        option[2]=k;
        return;
      }
    }
    for(var k in r) if (k!=x)
    {
      if (ar[k][y]==0)
      {
        option[1]=k;
        option[2]=y;
        return;
```

```
      }
    }
    fail();
  }



  write("[");


  for(var k in xyok)
  {
    var switchCell=0;
    var option=new Array(2);

    if (ar[k.i][k.j]==1) switchCell=1;
    if (switchCell==1)
    {

      lookforoption(k.i,k.j,option);
      apply(option[1],option[2]);
    }

    apply(k.i,k.j);
    if (switchCell==1)  apply(option[1],option[2]);

    if ((Opl.ord(xyok,k)+1) %20==0) writeln();
  }

  writeln("]");

}

// To double check that the job is done
assert forall(i,j in r) ar[i][j]==1;


/*

gives

[(1,24),(6,24),(7,24),(6,24),(9,24),(6,24),(11,24),(6,24),(12,24),(6,24),(13,24),(
6,24),(15,24),(18,24),(1,24),(19,24),(1,24),(20,24),(21,24),(23,24),(1,23),(2,23),
(3,23),(2,23),(5,23),(2,23),(11,23),(2,23),(13,23),(2,23),(14,23),(2,23),(15,23),(
2,23),(16,23),(2,23),
(17,23),(18,23),(1,23),(19,23),(1,23),(2,23),(22,23),(2,23),(1,23),(24,23),(1,23),
(1,22),(2,22),(3,22),(4,22),(1,22),(8,22),(1,22),(9,22),(11,22),(12,22),(13,22),(2
,22),(16,22),(2,22),(1,22),(18,22),(1,22),(2,22),(22,22),(2,22),(23,22),(24,22),(3
,21),(1,21),(3,21),
(1,21),(4,21),(1,21),(5,21),(1,21),(6,21),(1,21),(8,21),(11,21),(12,21),(1,21),(14
,21),(1,21),(15,21),(16,21),(18,21),(19,21),(3,21),(22,21),(3,21),(1,21),(24,21),(
1,21),(5,20),(1,20),(5,20),(2,20),(5,20),(1,20),(6,20),(1,20),(5,20),(9,20),(5,20)
,(1,20),(10,20),(1,20),(5,20),(11,20),(5,20),
(16,20),(5,20),(20,20),(5,20),(1,20),(21,20),(1,20),(5,20),(22,20),(5,20),(1,19),(
2,19),(4,19),(2,19),(1,19),(5,19),(1,19),(2,19),(7,19),(2,19),(8,19),(2,19),(12,19
),(2,19),(13,19),(17,19),(18,19),(2,19),(20,19),(2,19),(1,19),(21,19),(1,19),(2,19
),(22,19),(2,19),(23,19),(3,18),(2,18),(3,18),(1,18),(7,18),(1,18),(8,18),
(1,18),(10,18),(1,18),(3,18),(11,18),(3,18),(1,18),(12,18),(1,18),(13,18),(15,18),
(3,18),(21,18),(3,18),(1,18),(22,18),(1,18),(3,17),(1,17),(3,17),(2,17),(3,17),(4,
```

```
17),(5,17),(1,17),(10,17),(1,17),(12,17),(1,17),(13,17),(1,17),(14,17),(1,17),(16,
17),(1,17),(17,17),(18,17),(3,17),(19,17),(3,17),
(20,17),(22,17),(2,16),(3,16),(2,16),(10,16),(2,16),(11,16),(2,16),(13,16),(2,16),
(1,16),(18,16),(1,16),(20,16),(21,16),(2,16),(22,16),(2,16),(23,16),(24,16),(2,15)
,(1,15),(2,15),(1,15),(2,15),(1,15),(2,15),(3,15),(2,15),(4,15),(14,15),(1,15),(17
,15),(1,15),(18,15),
(19,15),(2,15),(21,15),(2,15),(1,15),(24,15),(1,15),(1,14),(2,14),(3,14),(2,14),(6
,14),(2,14),(12,14),(14,14),(1,14),(17,14),(1,14),(18,14),(19,14),(2,14),(20,14),(
2,14),(1,14),(22,14),(1,14),(24,14),(1,13),(2,13),(3,13),(4,13),(6,13),
(2,13),(7,13),(2,13),(10,13),(2,13),(11,13),(2,13),(1,13),(14,13),(1,13),(17,13),(
1,13),(18,13),(1,13),(2,13),(20,13),(2,13),(1,13),(24,13),(1,13),(1,12),(4,12),(5,
12),(3,12),(7,12),(3,12),(1,12),(8,12),(1,12),(9,12),(1,12),(12,12),(1,12),(3,12),
(15,12),(3,12),(1,12),(17,12),(1,12),(3,12),(22,12),(3,12),(1,12),(23,12),(1,12),(
24,12),
(3,11),(4,11),(3,11),(1,11),(5,11),(1,11),(3,11),(6,11),(3,11),(10,11),(3,11),(12,
11),(3,11),(1,11),(14,11),(1,11),(15,11),(23,11),(3,11),(24,11),(3,11),(1,10),(2,1
0),(1,10),(7,10),(9,10),(12,10),(14,10),(19,10),(21,10),(24,10),(4,9),(1,9),(4,9),
(6,9),(4,9),(10,9),(4,9),
(1,9),(11,9),(1,9),(12,9),(1,9),(19,9),(1,9),(2,8),(1,8),(2,8),(1,8),(4,8),(1,8),(
2,8),(5,8),(2,8),(6,8),(2,8),(10,8),(2,8),(1,8),(11,8),(1,8),(2,8),(16,8),(2,8),(1
,7),(2,7),(1,7),(3,7),(1,7),(4,7),(5,7),(7,7),(10,7),(2,7),(12,7),(2,7),(13,7),(15
,7),
(19,7),(2,7),(21,7),(2,7),(22,7),(3,6),(2,6),(3,6),(4,6),(3,6),(6,6),(3,6),(7,6),(
3,6),(9,6),(3,6),(15,6),(3,6),(16,6),(3,6),(1,6),(19,6),(1,6),(20,6),(1,6),(21,6),
(1,6),(3,6),(23,6),(3,6),(1,6),(24,6),(1,6),(6,5),(1,5),(6,5),(2,5),(6,5),(4,5),(6
,5),(5,5),(6,5),(7,5),(6,5),
(11,5),(6,5),(12,5),(6,5),(1,5),(14,5),(1,5),(6,5),(15,5),(6,5),(1,5),(18,5),(1,5)
,(19,5),(22,5),(6,5),(23,5),(6,5),(1,4),(2,4),(3,4),(2,4),(4,4),(2,4),(5,4),(2,4),
(1,4),(8,4),(1,4),(10,4),(11,4),(2,4),(15,4),(2,4),(1,4),(17,4),(1,4),(18,4),(20,4
),(2,4),(23,4),(2,4),
(24,4),(3,3),(1,3),(3,3),(6,3),(3,3),(9,3),(3,3),(1,3),(10,3),(1,3),(3,3),(11,3),(
3,3),(1,3),(12,3),(1,3),(3,3),(13,3),(3,3),(15,3),(17,3),(1,3),(18,3),(1,3),(3,3),
(19,3),(3,3),(23,3),(3,3),(24,3),(3,3),(2,2),(3,2),(2,2),(4,2),(2,2),(5,2),(8,2),(
1,2),(11,2),(1,2),
(13,2),(14,2),(15,2),(16,2),(2,2),(20,2),(2,2),(1,2),(22,2),(1,2),(23,2),(1,1),(3,
24),(3,1),(3,24),(4,1),(5,24),(5,1),(5,24),(6,1),(7,24),(7,1),(7,24),(9,1),(14,24)
,(14,1),(14,24),(16,1),(19,24),(19,1),(19,24),(20,1),(21,24),(21,1),(21,24),(24,1)
,
]


*/
```