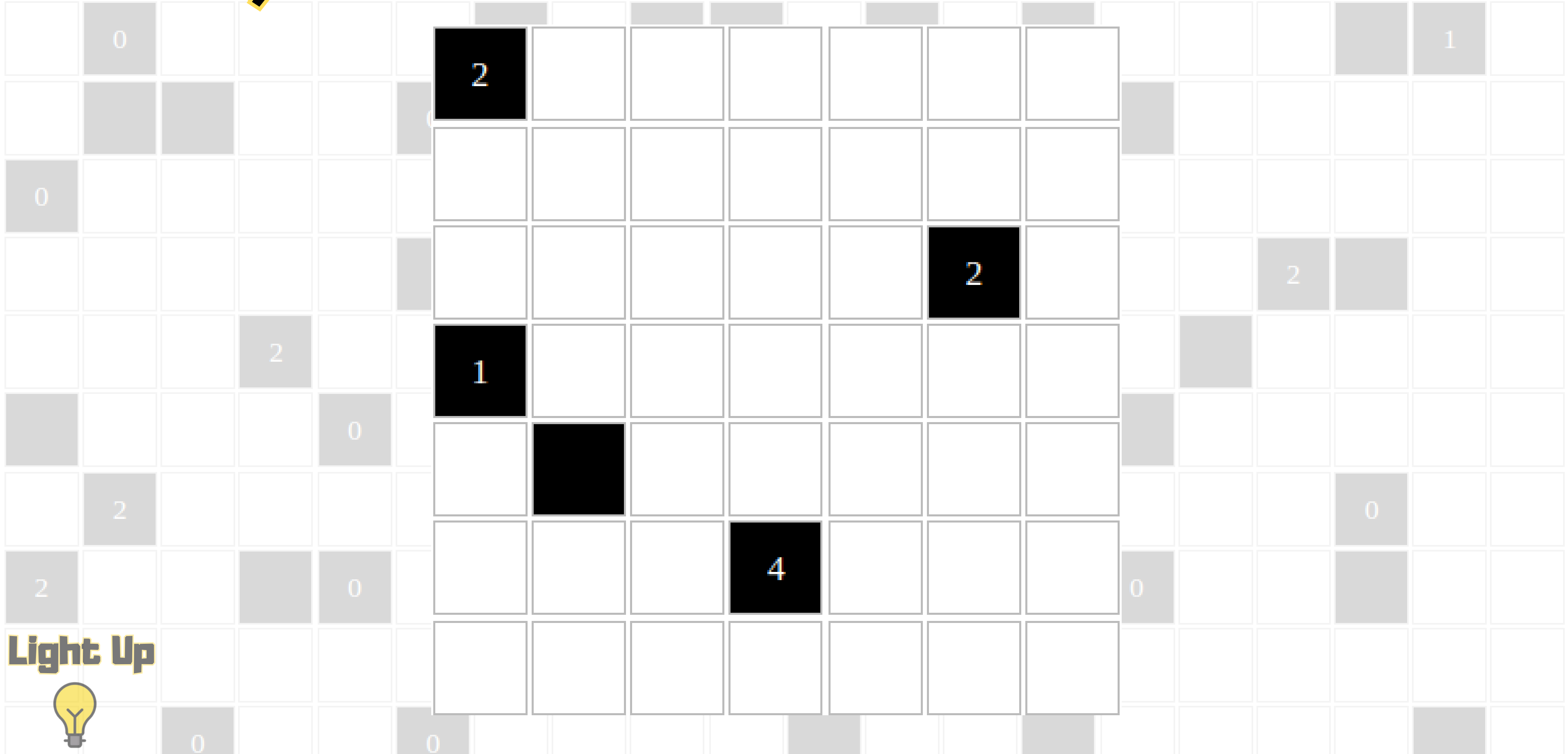


# Light Up



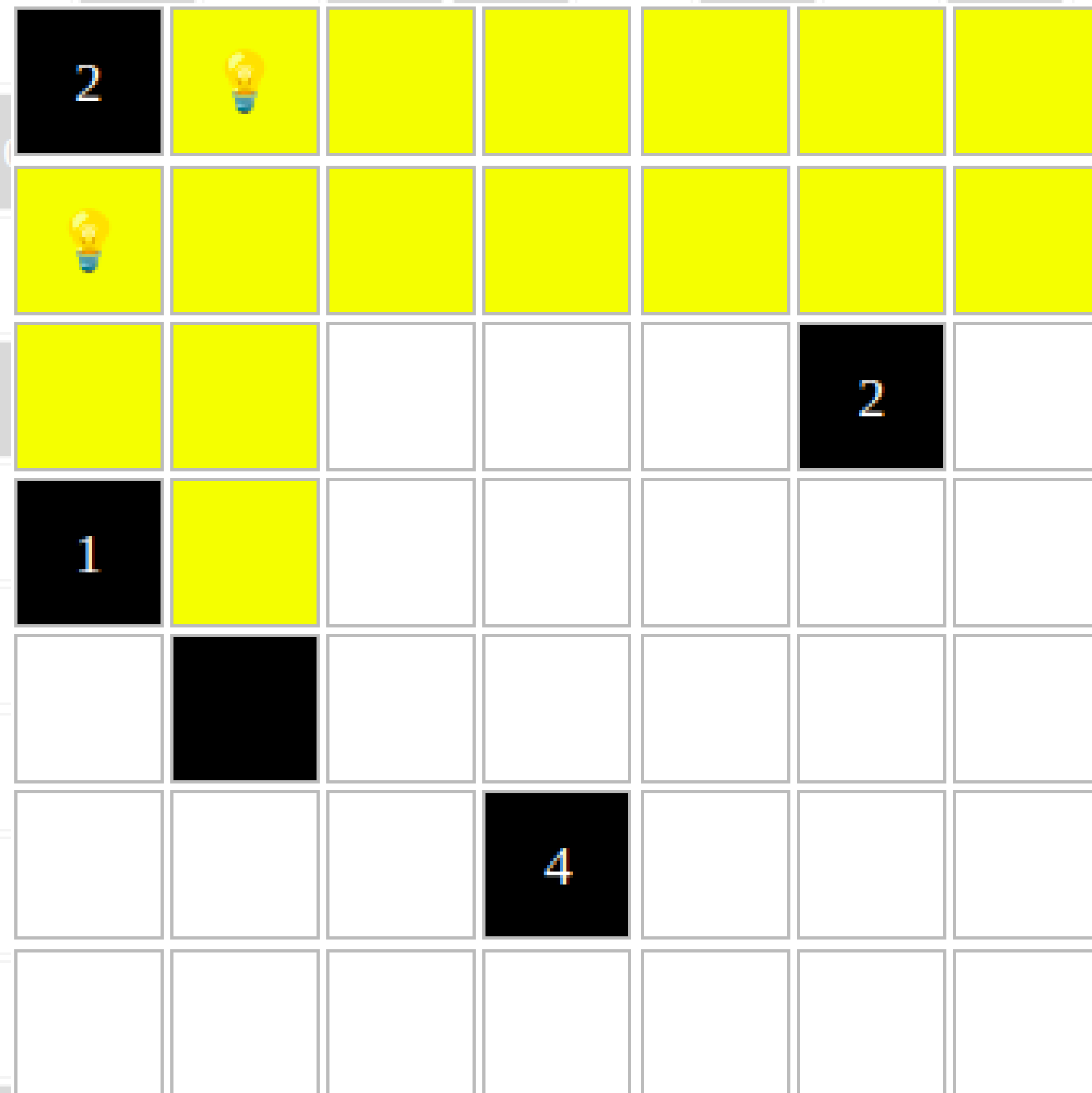
# Solução - Trivial



Light Up



# Solução - Trivial



Light Up



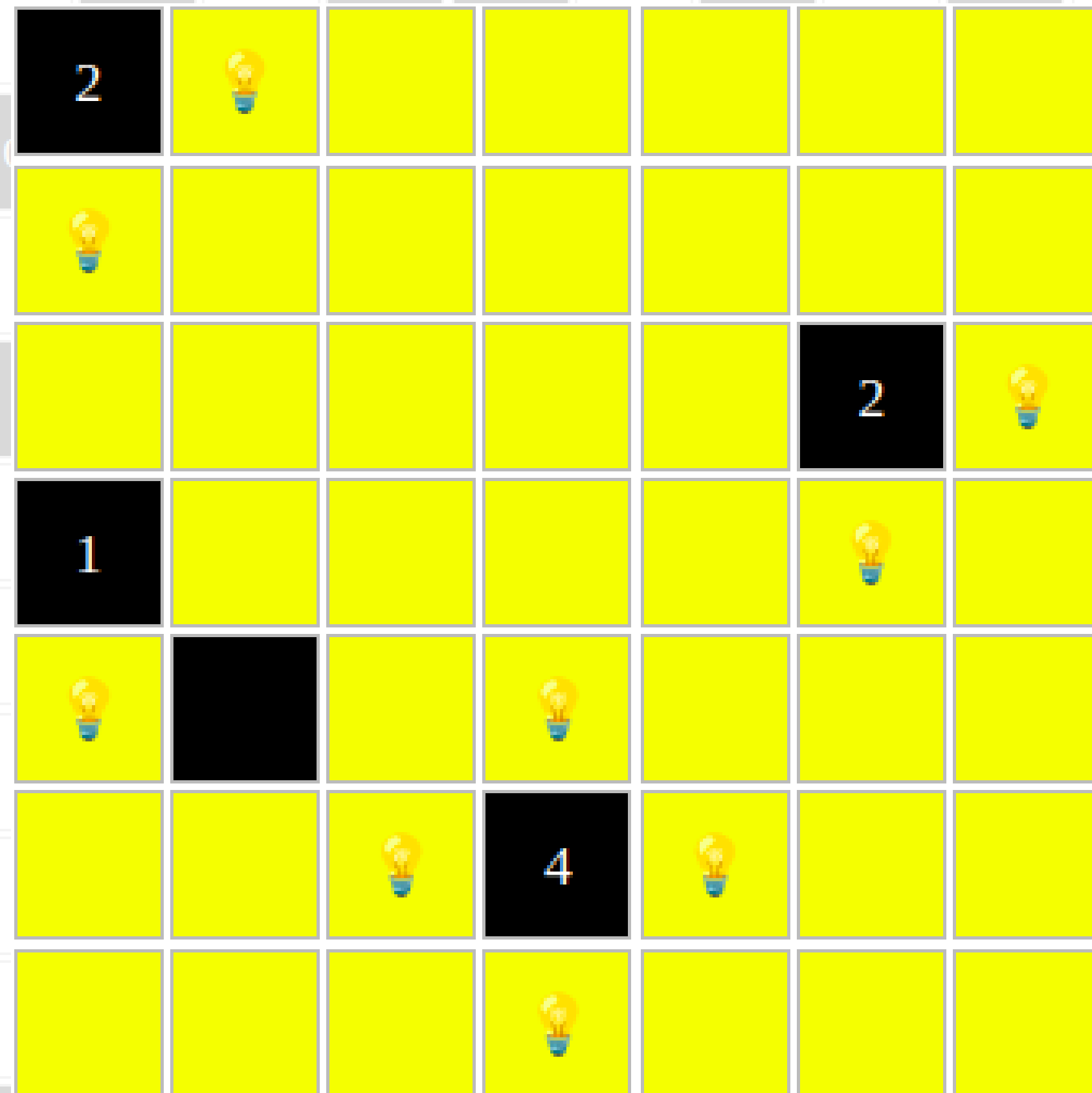
# Solução - Trivial

2	💡					
💡						
					2	
1						
			💡			
		💡	4	💡		
			💡			

Light Up



# Solução - Trivial



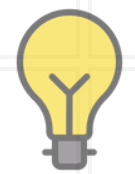
Light Up



# Representação do grid

(0,0)	(0,1)		(0,3)	(0,4)	(0,5)	(0,6)
(1,0)	1	(1,2)	(1,3)	(1,4)	3	(1,6)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
2	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,0)	3	(5,2)	(5,3)	(5,4)	1	(5,6)
(6,0)	(6,1)	(6,2)	(6,3)	0	(6,5)	(6,6)

Light Up



# Identificação das linhas

1	1		2	2	2	2
3	1	4	4	4	3	5
6	6	6	6	6	6	
7	7	7	7	7	7	7
2	8	8	8	8	8	8
9	3	10	10	10	1	11
12	12	12	12	0	13	13

Light Up



# Identificação das colunas

1	2		6	7	8	11
1	1	5	6	7	3	11
1	3	5	6	7	9	
1	3	5	6	7	9	12
2	3	5	6	7	9	12
1	3	5	6	7	1	12
1	4	5	6		10	12

Light Up





# Predicados

```
(:types
  cell line column - object
)

(:predicates
  (line_iluminated ?l - line)
  (column_iluminated ?col - column)
  (cell_in_line ?c - cell ?l - line)
  (cell_in_column ?c - cell ?col - column)
)
```

Light Up



# Action

```
(:action bulb
  :parameters (?c - cell ?l - line ?col - column)
  :precondition (and
    (cell_in_line ?c ?l)
    (cell_in_column ?c ?col)
    (not (line_illuminated ?l))
    (not (column_illuminated ?col))
  )
  :effect (and
    (line_illuminated ?l)
    (column_illuminated ?col)
  )
)
```

Light Up



# Melhoria - merge

(1,1)	(1,2)		(2,6)	(2,7)	(2,8)	(2,11)
(3,1)	1	(4,5)	(4,6)	(4,7)	3	(5,11)
(6,1)	(6,3)	(6,5)	(6,6)	(6,7)	(6,9)	
(7,1)	(7,3)	(7,5)	(7,6)	(7,7)	(7,9)	(7,12)
2	(8,3)	(8,5)	(8,6)	(8,7)	(8,9)	(8,12)
(9,1)	3	(10,5)	(10,6)	(10,7)	1	(11,12)
(12,1)	(12,4)	(12,5)	(12,6)		(13,10)	(13,12)

Light Up



```
(:types
  line column - object
)

(:predicates
  (line_illuminated ?l - line)
  (column_illuminated ?col - column)
  (cell ?l - line ?col - column)
)

(:action bulb
  :parameters (?l - line ?col - column)
  :precondition (and
    (cell ?l ?col)
    (not (line_illuminated ?l))
    (not (column_illuminated ?col))
  )
  :effect (and
    (line_illuminated ?l)
    (column_illuminated ?col)
  )
)
```

**Light Up**



# Prós e contras

- **Menos parâmetros enumerados**
- **Menos tipos**
- **Menos predicados a serem inicializados**
- **Dificuldade em traduzir quais células foram acesas**  
**[necessidade de um tradutor externo]**
- **Necessidade adicional de verificar se a célula é válida** (o par linha e coluna existe)

**Conclusão: vale a pena, mas seria consideravelmente melhor caso acertasse apenas as células válidas**

Light Up



# Células pretas

**Células que exigem uma quantidade fixa de lâmpadas adjacentes**

- **Contadores**

- **Resolvem o problema**
- **São mais genéricos**
- **Mais complexo para codar**
- **Custo computacional aumenta consideravelmente**

- **Verificações estáticas - “cheat”**

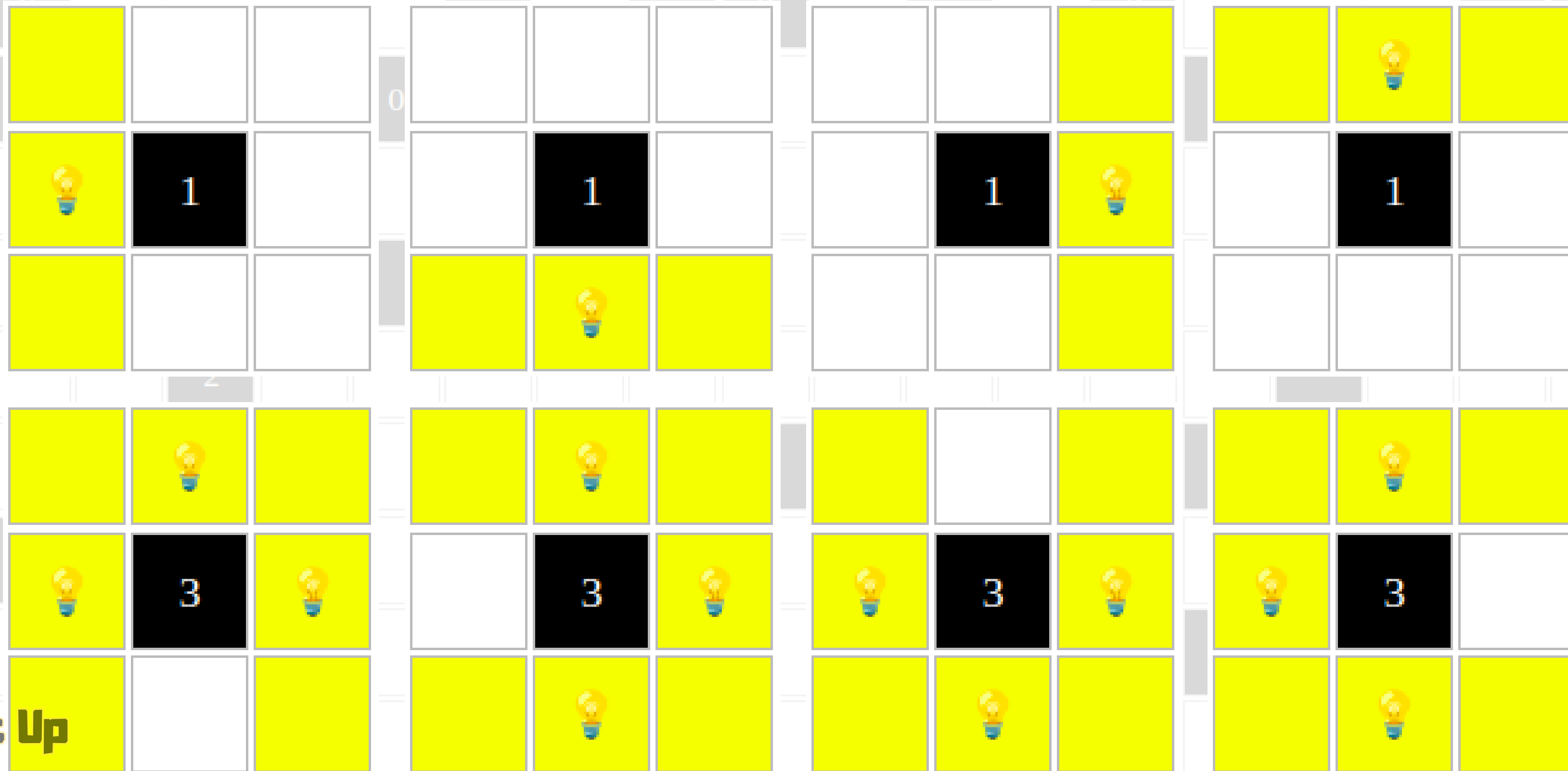
- **Resolvem o problema**
- **São menos genéricos**
- **Fácil para codar**
- **Pode dar problema em diferentes planejadores**

- **Custo computacional - ou muito caro, ou muito barato (problema)**

Light Up



# Verificações estáticas



Light Up



# Verificações estáticas

```
(:goal (and
  (grid_illuminated)
  (or
    (and
      (bulb l_1 c_2)
      (not (bulb l_2 c_1))
      (not (bulb l_3 c_4))
      (not (bulb l_4 c_3))
    )
    (and
      (not (bulb l_1 c_2))
      (bulb l_2 c_1)
      (not (bulb l_3 c_4))
      (not (bulb l_4 c_3))
    )
  )
  ...
)
```

Light Up





# Matriz iluminada

- Uma célula possui um par  $(L, C)$ 
  - Se sua linha ou sua coluna estiver acesa, a célula está acesa

```
(:action grid_iluminated
  :parameters ()
  :precondition (and
    (forall (?l - line ?col - column)
      (or
        (not (cell ?l ?col))
        (line_iluminated ?l)
        (column_iluminated ?col)
      )
    )
  )
  :effect (and (grid_iluminated))
)
```

Light Up



# Matriz iluminada

- **Essa estratégia é lenta**
  - **Realiza uma série de enumerações desnecessárias**
- **Podemos repetir a ideia da verificação estática**

```
(:action grid_iluminated
  :parameters ()
  :precondition (and
    (or
      (line_iluminated l_1)
      (column_iluminated c_1)
    )
    (or
      (line_iluminated l_1)
      ...
      ...
    )
  )
  :effect (and (grid_iluminated))
)
```

Light Up



# Matriz iluminada

- Há uma maneira de fazer menos verificações ainda
- Uma linha está iluminada caso o predicado seja verdadeiro ou o conjunto de colunas que fazem parte da linha está iluminado
- A mesma equivalência pode valer para as colunas
- Escolho a menor quantidade entre linhas e colunas

(1,1)	(1,2)		(2,6)	(2,7)	(2,8)	(2,11)
(3,1)	1	(4,5)	(4,6)	(4,7)	3	(5,11)
(6,1)	(6,3)	(6,5)	(6,6)	(6,7)	(6,9)	

Light Up



# Considerações

- **51/51 mapas resolvidos na track ágil com o score de ~38.763**
- **Era muito bom antes da adição dos 20 mapas**
- **Ainda se atrapalha por conta da enumeração de células que não são válidas**
- **A verificação de células pretas também pode ser melhorada**

Light Up



# Chegando no 51

- **Problema: enumeração de células não válidas**
  - **Resolução: uma action para cada célula**
  - **Vantagem: podemos se livrar da tradução externa**

```
(:action bulb_x_y
  :parameters ()
  :precondition (and
    (not (line_iluminated l_{l(x)}))
    (not (column_iluminated c_{c(x)}))
  )
  :effect (and
    (line_iluminated l_{l(x)})
    (column_iluminated c_{c(x)})
    (bulb c_x_y)
  )
)
```

Light Up



# Chegando no 51

- Agora, vale mais a pena a utilização de contadores
- A utilização do “when” mostrou ser lenta
- A adição de actions para o contador também mostrou ser lenta
- A solução foi fazer actions com precondições que substitui o uso do “when”
  - A dificuldade é quando uma célula possui adjacência com mais de uma célula<sup>2</sup> preta
  - É necessário fazer todas as permutações desse contador

Light Up



# Chegando no 51

```
(:action bulb_x_y_2
  :parameters ()
  :precondition (and
    (not (line_iluminated l_{l(x)}))
    (not (column_iluminated c_{c(x)}))
    (black_{id}_2)
  )
  :effect (and
    (line_iluminated l_{l(x)})
    (column_iluminated c_{c(x)})
    (not (black_{id}_2))
    (black_{id}_1)
  )
)
```

Light Up



# Chegando no 51

```
(:action bulb_x_y_1
  :parameters ()
  :precondition (and
    (not (line_iluminated l_{l(x)}))
    (not (column_iluminated c_{c(x)}))
    (black_{id}_1)
  )
  :effect (and
    (line_iluminated l_{l(x)})
    (column_iluminated c_{c(x)})
    (not (black_{id}_1))
    (black_{id}_0)
  )
)
```

Light Up





