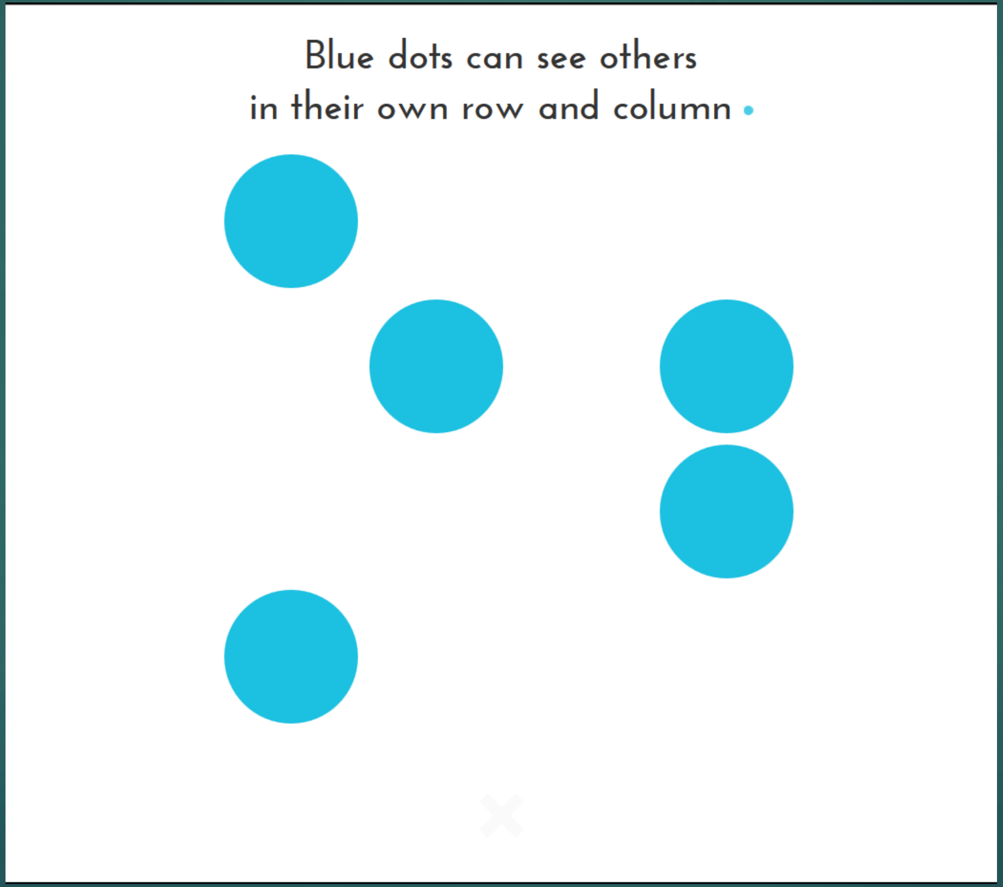
Sistemas de Inteligencia Artificial.

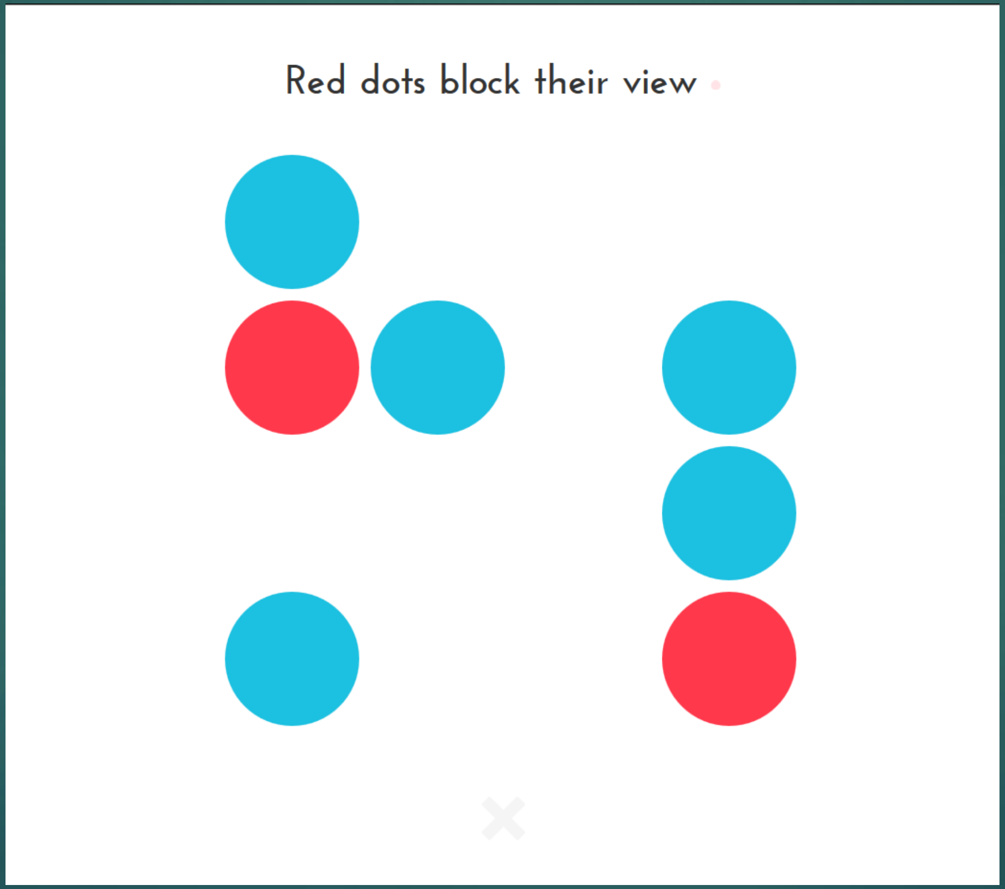
METODOS DE BUSQUEDA NO INFORMADOS E INFORMADOS

Problema – 0h n0



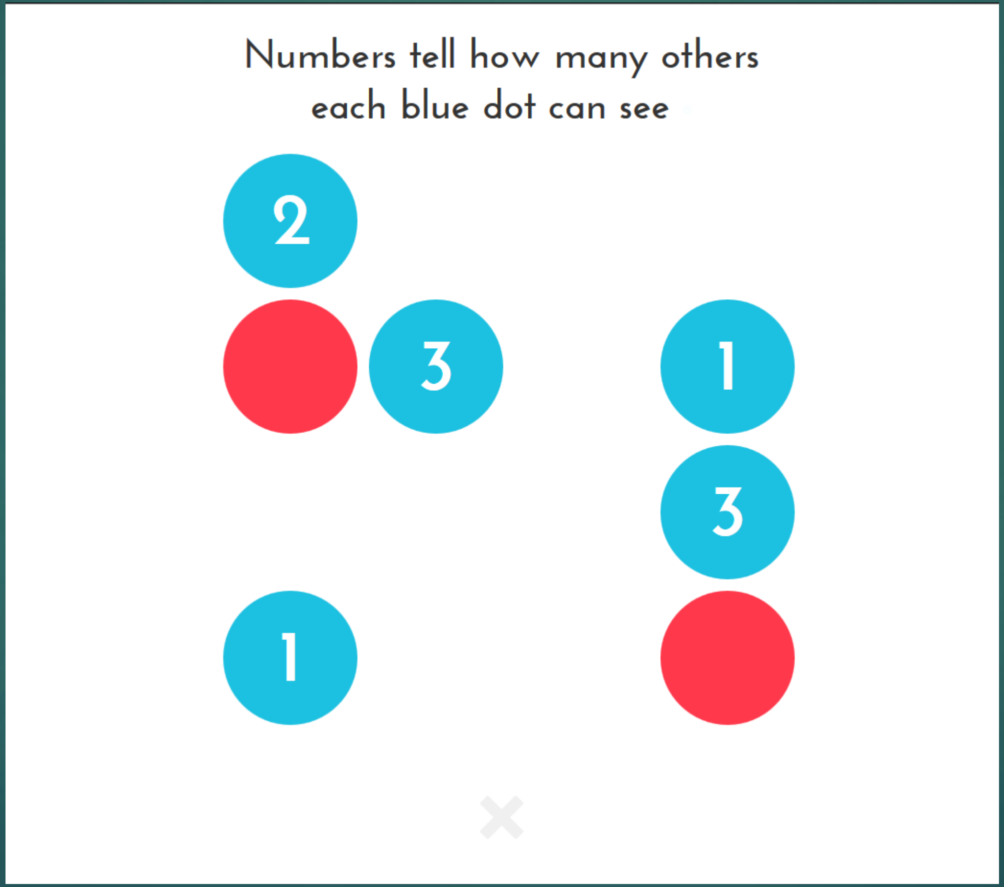
Blue dots can see others
in their own row and column •

Problema – 0h n0



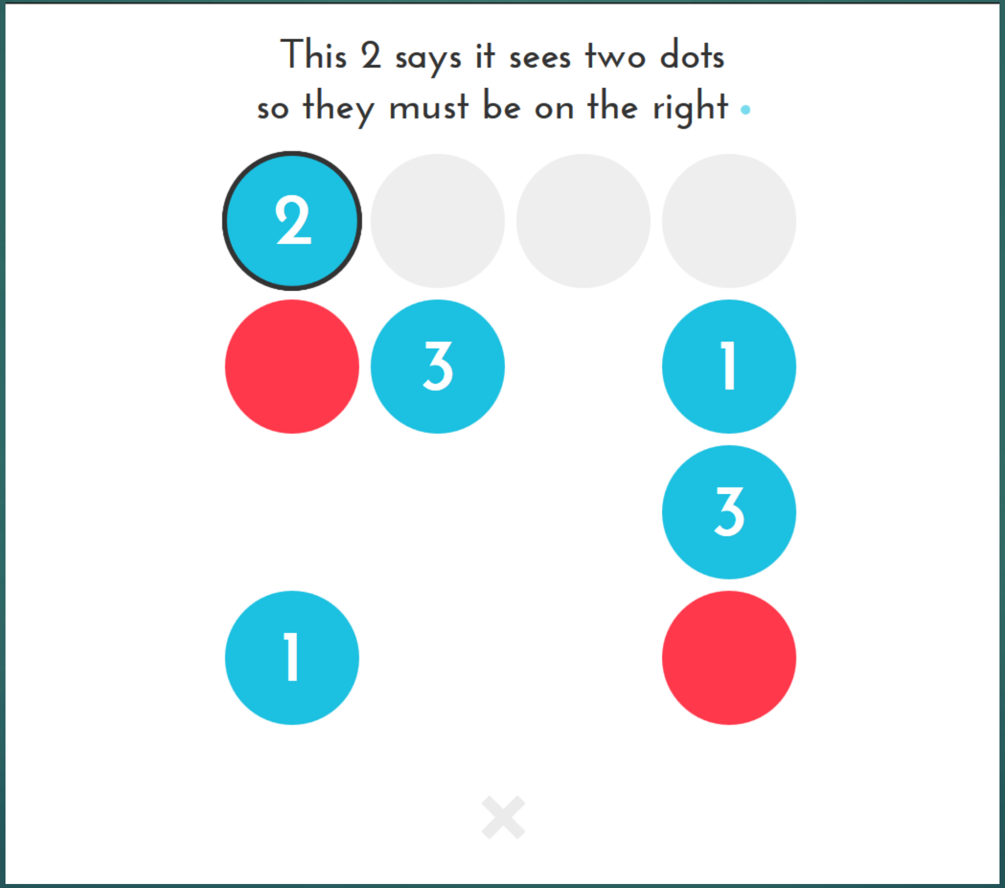
Red dots block their view

Problema – 0h n0



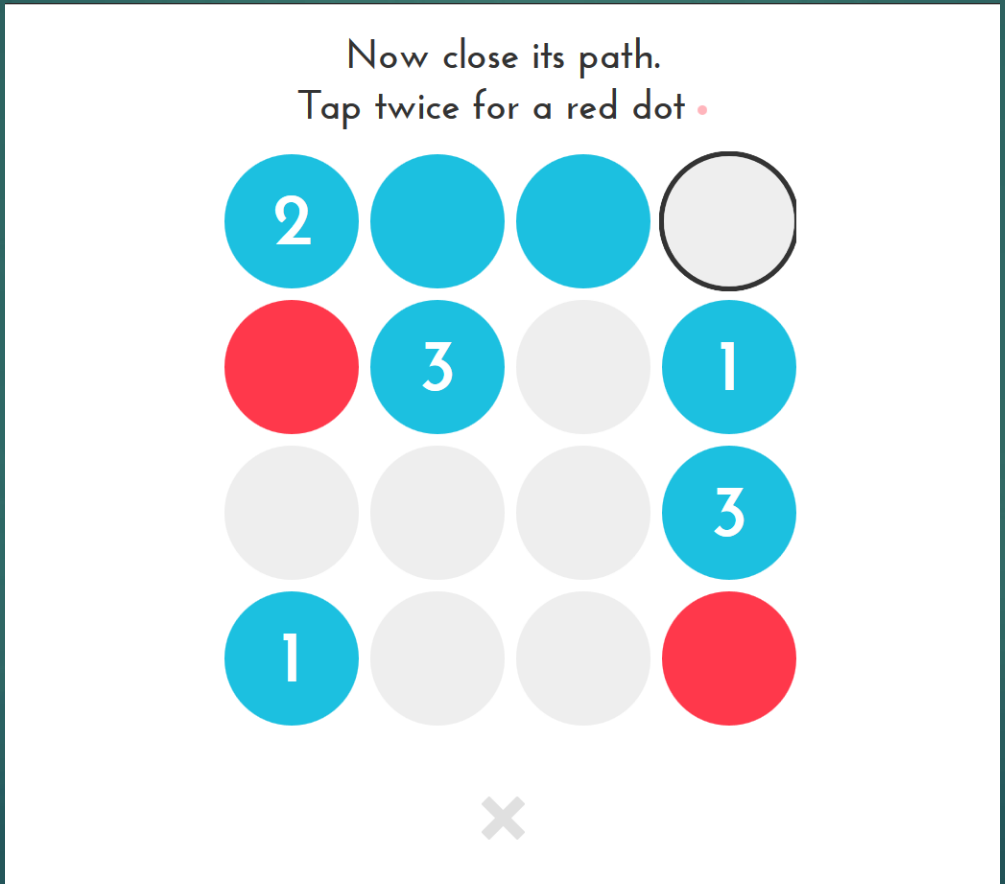
Numbers tell how many others each blue dot can see

Problema – 0h n0



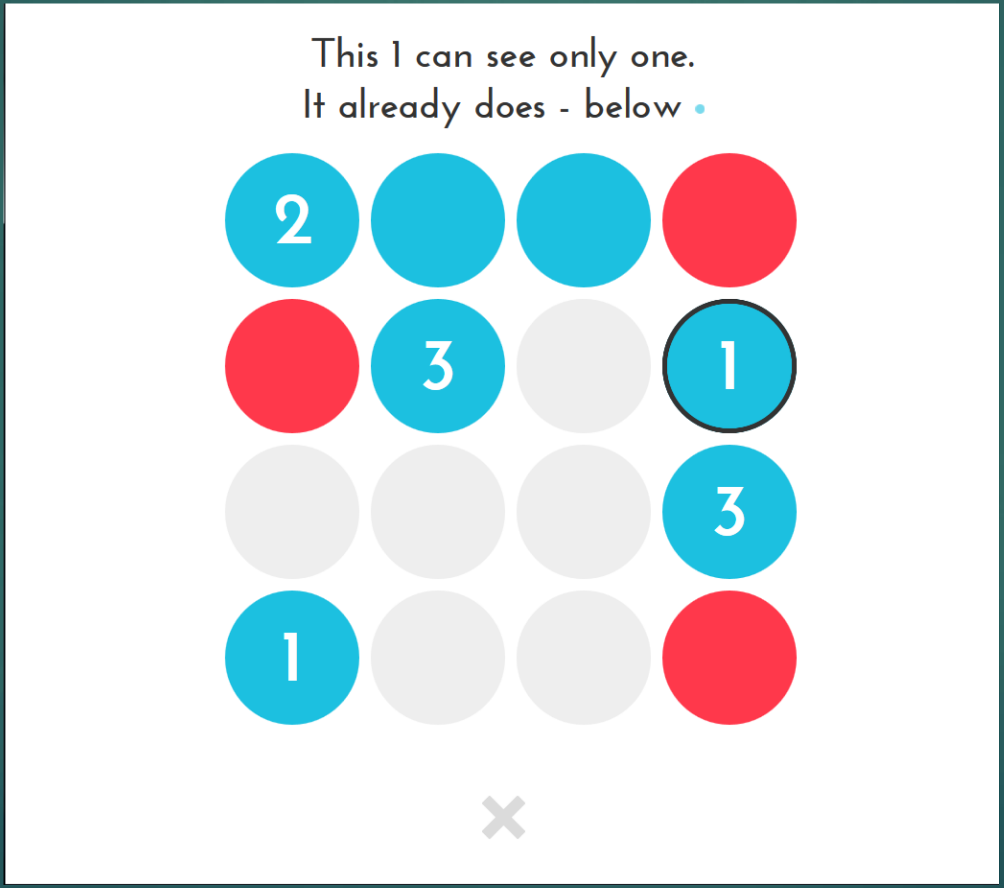
This 2 says it sees two dots
so they must be on the right

Problema – 0h n0



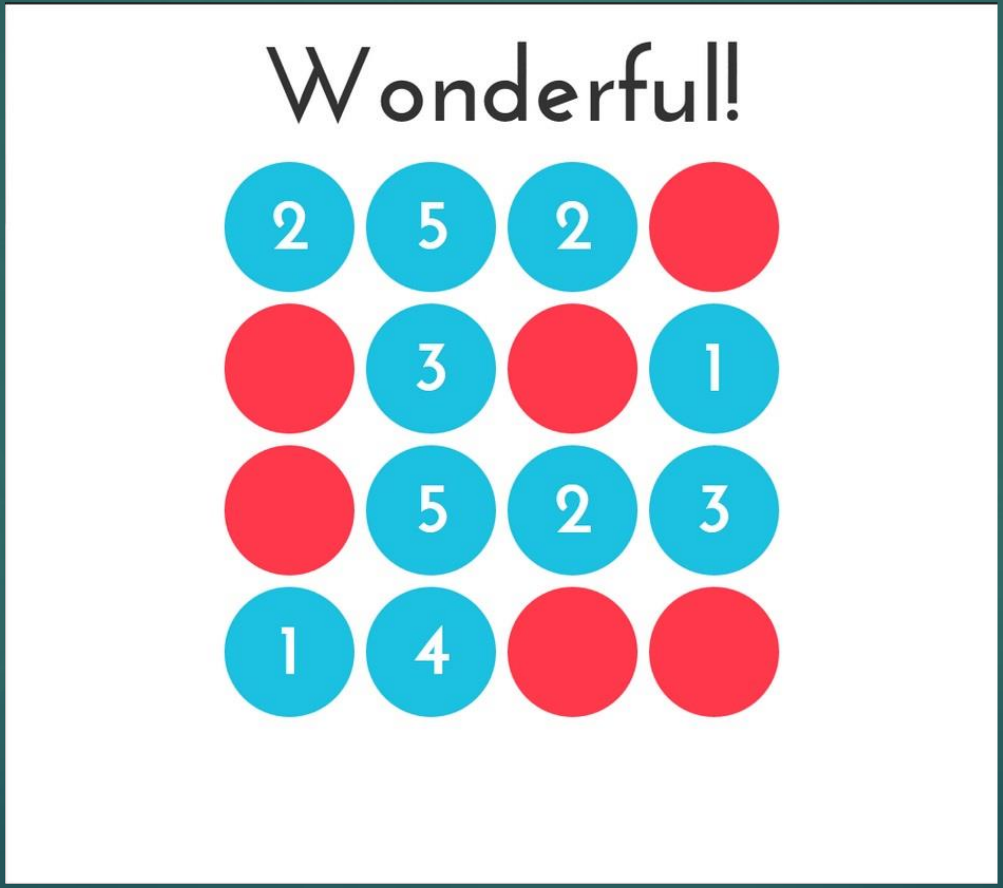
Now close its path.
Tap twice for a red dot •

Problema – 0h n0



This 1 can see only one.
It already does - below

Problema – 0h n0



Wonderful!

2	5	2	🟥
🟥	3	🟥	1
🟥	5	2	3
1	4	🟥	🟥

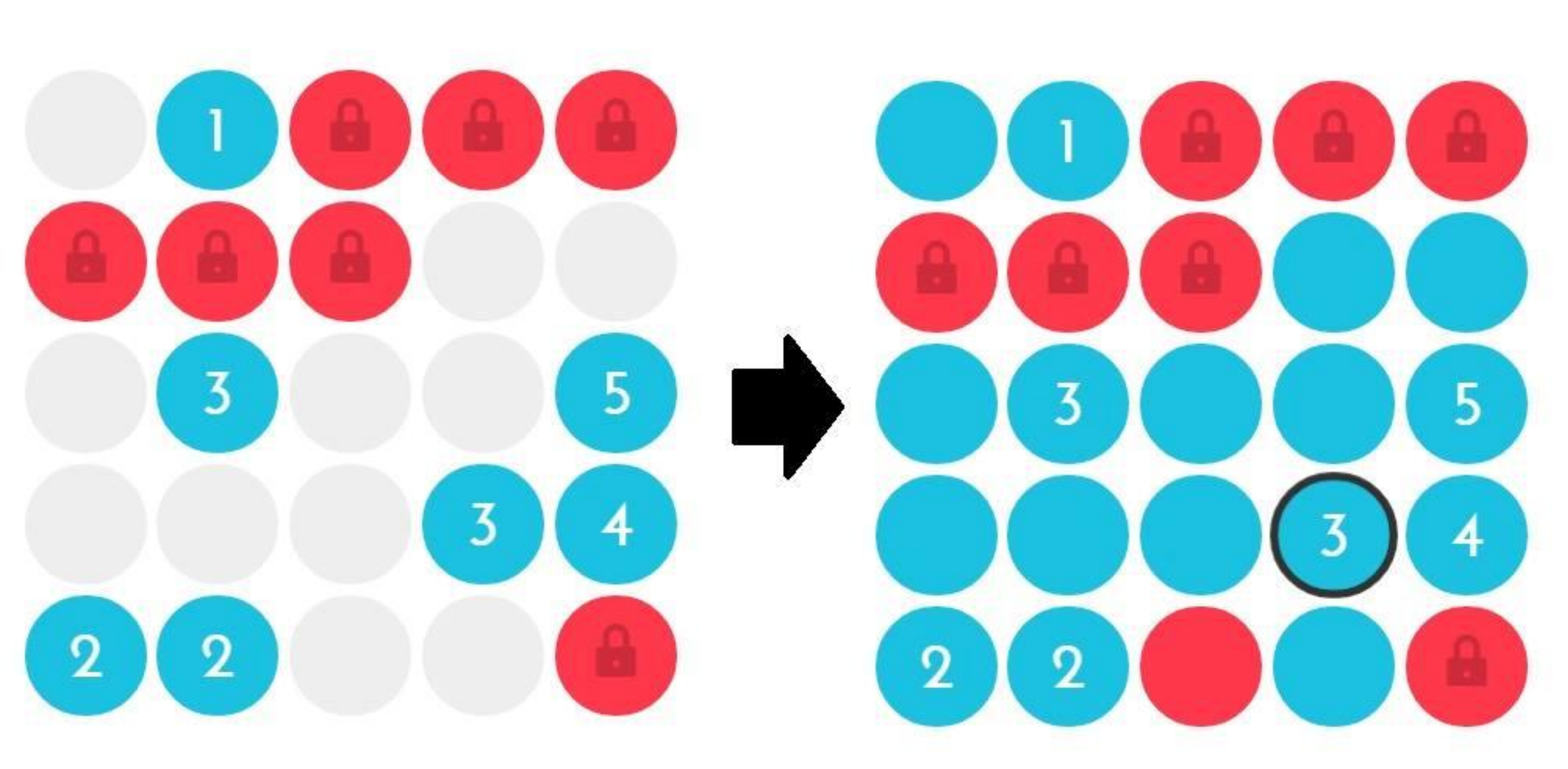
Heurísticas

Primera Heurística

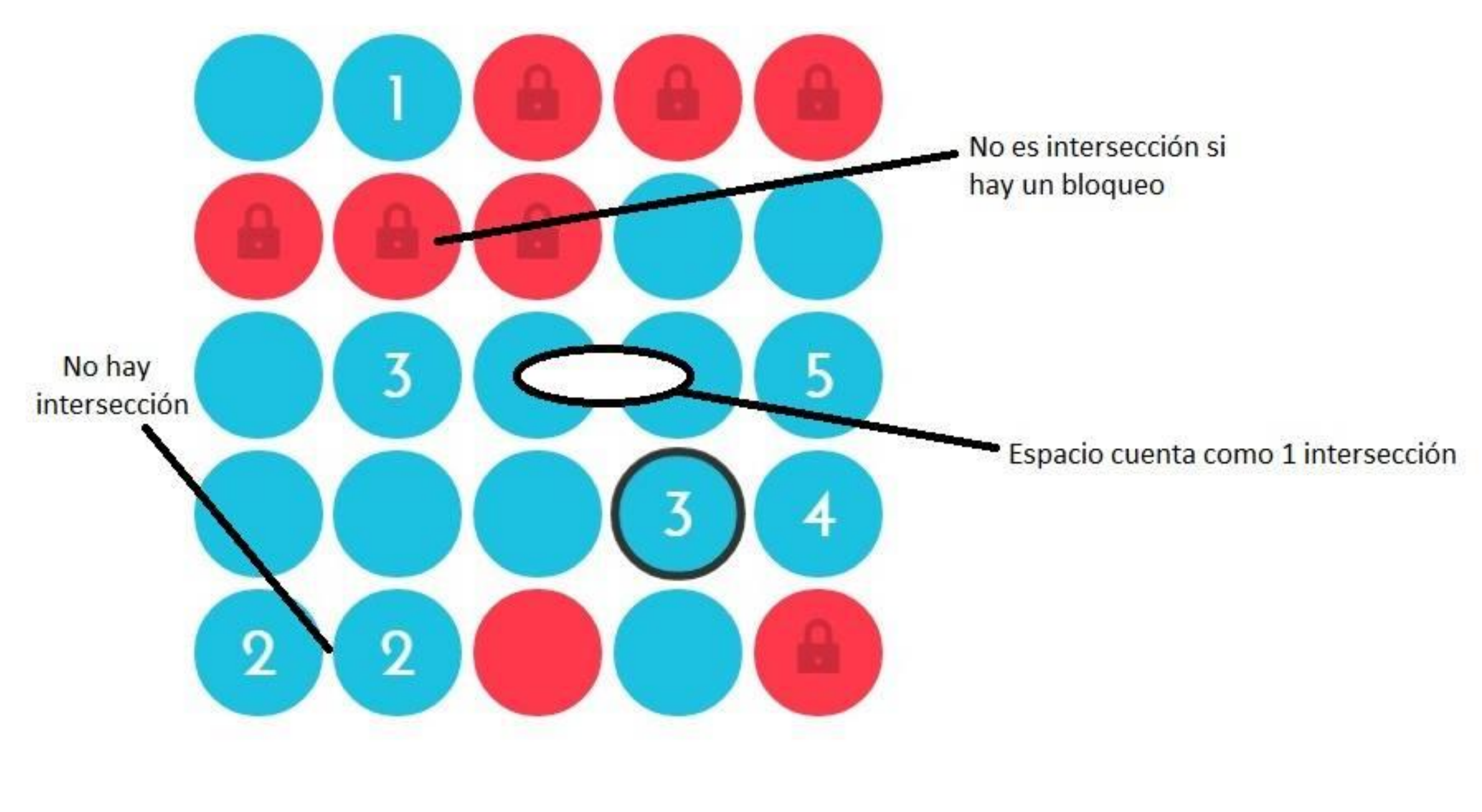
Se tiene en cuenta:

- Cantidad de celdas sin satisfacer (n)
- Cantidad de intersecciones (i)
- $h = n - i - 1$

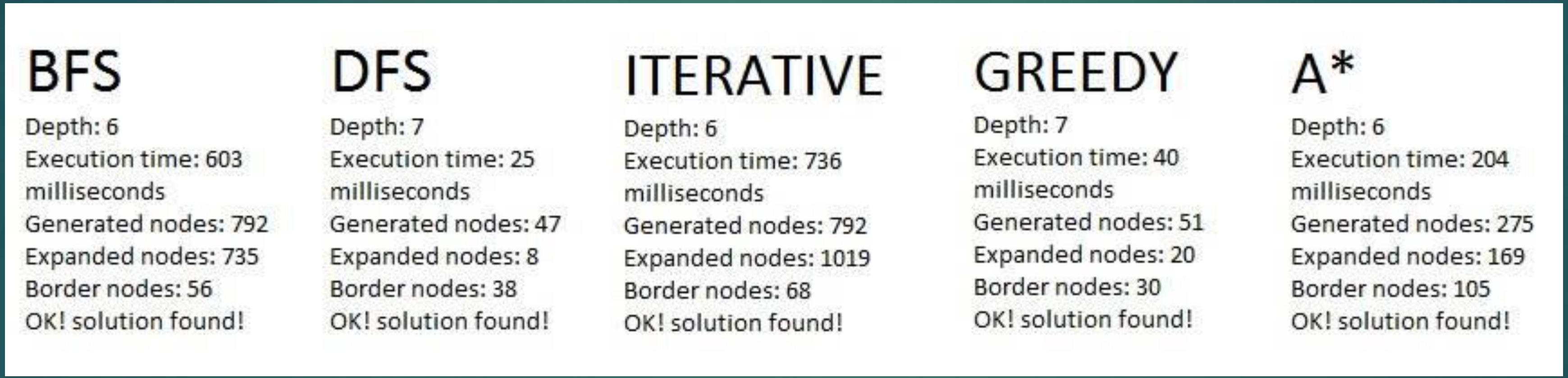
Primera Heurística – Preparación



Primera Heurística - Cálculo


1
No es intersección si hay un bloqueo
No hay intersección
3
5
Espacio cuenta como 1 intersección
3
4
2
2

Primera Heurística - Resultados



BFS

Depth: 6
Execution time: 603 milliseconds
Generated nodes: 792
Expanded nodes: 735
Border nodes: 56
OK! solution found!

DFS

Depth: 7
Execution time: 25 milliseconds
Generated nodes: 47
Expanded nodes: 8
Border nodes: 38
OK! solution found!

ITERATIVE

Depth: 6
Execution time: 736 milliseconds
Generated nodes: 792
Expanded nodes: 1019
Border nodes: 68
OK! solution found!

GREEDY

Depth: 7
Execution time: 40 milliseconds
Generated nodes: 51
Expanded nodes: 20
Border nodes: 30
OK! solution found!

A*

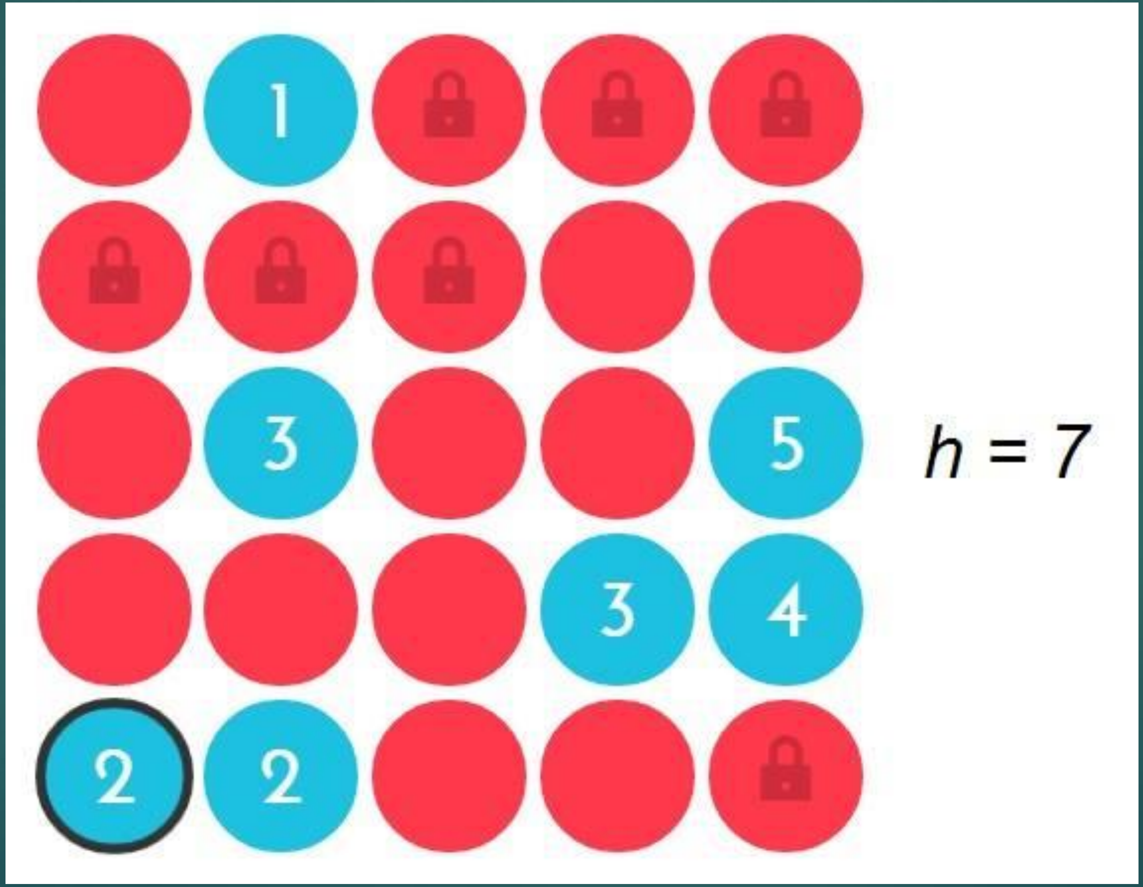
Depth: 6
Execution time: 204 milliseconds
Generated nodes: 275
Expanded nodes: 169
Border nodes: 105
OK! solution found!

Segunda Heurística

Se tiene en cuenta:

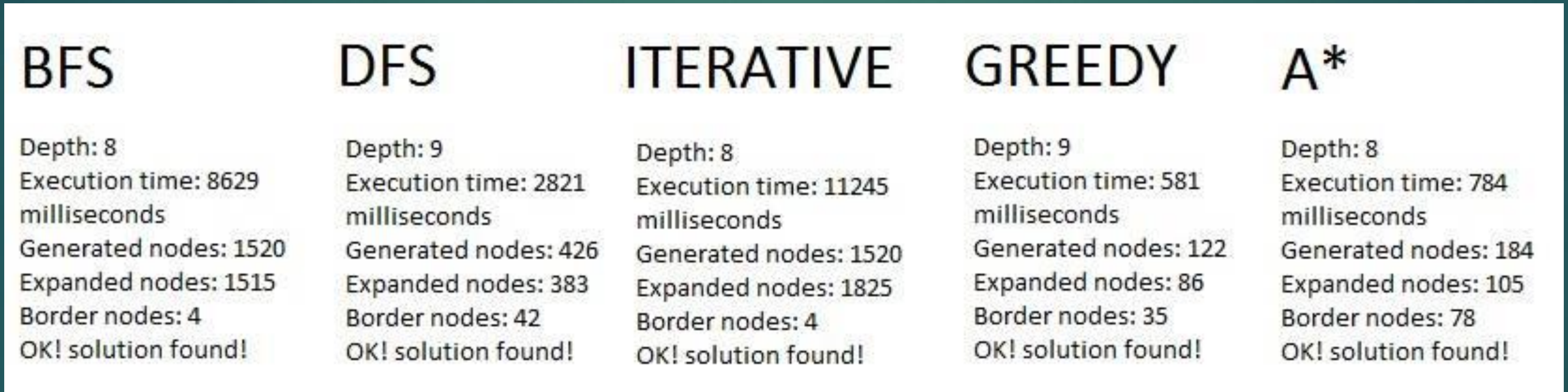
- Cantidad de nodos sin satisfacer
- Se prepara el tablero con fichas rojas
- No afectan las intersecciones

Segunda Heurística - Cálculo



$h = 7$

Segunda Heurística - Resultados



BFS

```
Depth: 8
Execution time: 8629
milliseconds
Generated nodes: 1520
Expanded nodes: 1515
Border nodes: 4
OK! solution found!
```

DFS

```
Depth: 9
Execution time: 2821
milliseconds
Generated nodes: 426
Expanded nodes: 383
Border nodes: 42
OK! solution found!
```

ITERATIVE

```
Depth: 8
Execution time: 11245
milliseconds
Generated nodes: 1520
Expanded nodes: 1825
Border nodes: 4
OK! solution found!
```

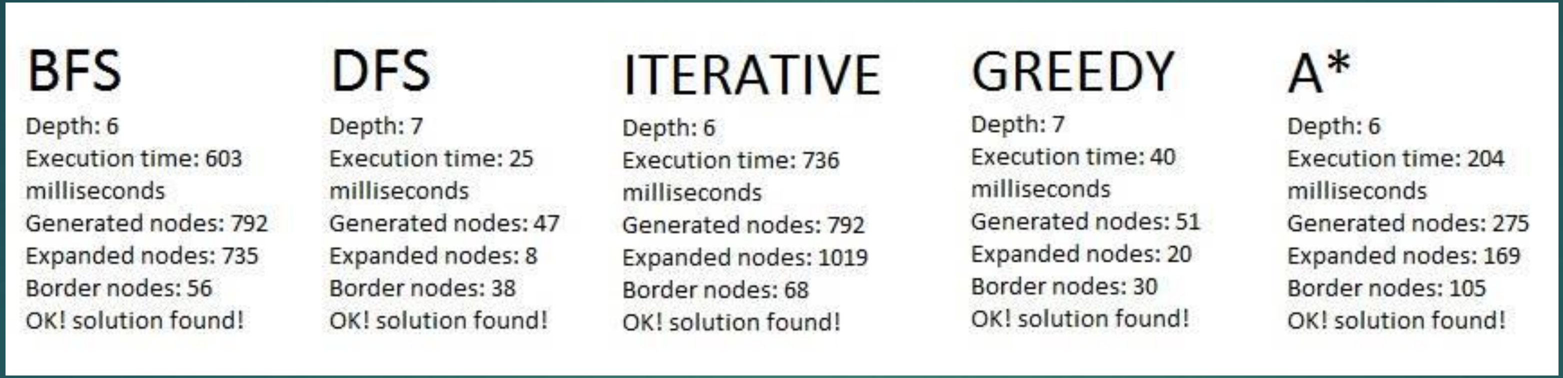
GREEDY

```
Depth: 9
Execution time: 581
milliseconds
Generated nodes: 122
Expanded nodes: 86
Border nodes: 35
OK! solution found!
```

A*

```
Depth: 8
Execution time: 784
milliseconds
Generated nodes: 184
Expanded nodes: 105
Border nodes: 78
OK! solution found!
```

Resultados - Comparación



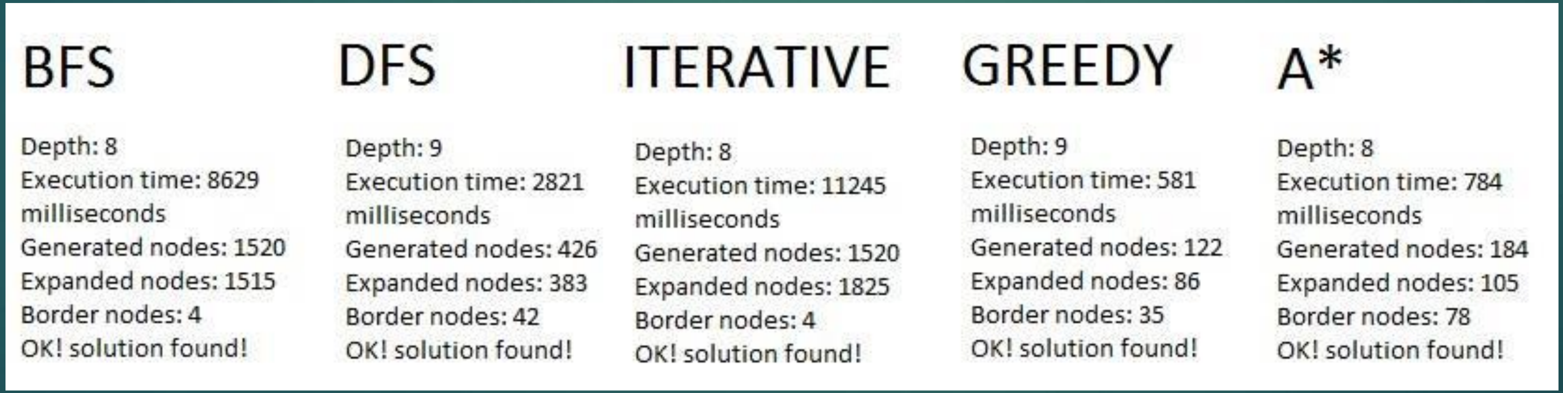
```
BFS
Depth: 6
Execution time: 603
milliseconds
Generated nodes: 792
Expanded nodes: 735
Border nodes: 56
OK! solution found!

DFS
Depth: 7
Execution time: 25
milliseconds
Generated nodes: 47
Expanded nodes: 8
Border nodes: 38
OK! solution found!

ITERATIVE
Depth: 6
Execution time: 736
milliseconds
Generated nodes: 792
Expanded nodes: 1019
Border nodes: 68
OK! solution found!

GREEDY
Depth: 7
Execution time: 40
milliseconds
Generated nodes: 51
Expanded nodes: 20
Border nodes: 30
OK! solution found!

A*
Depth: 6
Execution time: 204
milliseconds
Generated nodes: 275
Expanded nodes: 169
Border nodes: 105
OK! solution found!
```



```
BFS
Depth: 8
Execution time: 8629
milliseconds
Generated nodes: 1520
Expanded nodes: 1515
Border nodes: 4
OK! solution found!

DFS
Depth: 9
Execution time: 2821
milliseconds
Generated nodes: 426
Expanded nodes: 383
Border nodes: 42
OK! solution found!

ITERATIVE
Depth: 8
Execution time: 11245
milliseconds
Generated nodes: 1520
Expanded nodes: 1825
Border nodes: 4
OK! solution found!

GREEDY
Depth: 9
Execution time: 581
milliseconds
Generated nodes: 122
Expanded nodes: 86
Border nodes: 35
OK! solution found!

A*
Depth: 8
Execution time: 784
milliseconds
Generated nodes: 184
Expanded nodes: 105
Border nodes: 78
OK! solution found!
```

Conclusiones

- Menos nodos explotados en Búsquedas Informadas
- El método Greedy fue el más eficiente para nuestro problema
- El DFS obtuvo mejores métricas debido al tablero elegido
  - Solución ubicada en la primer rama del árbol