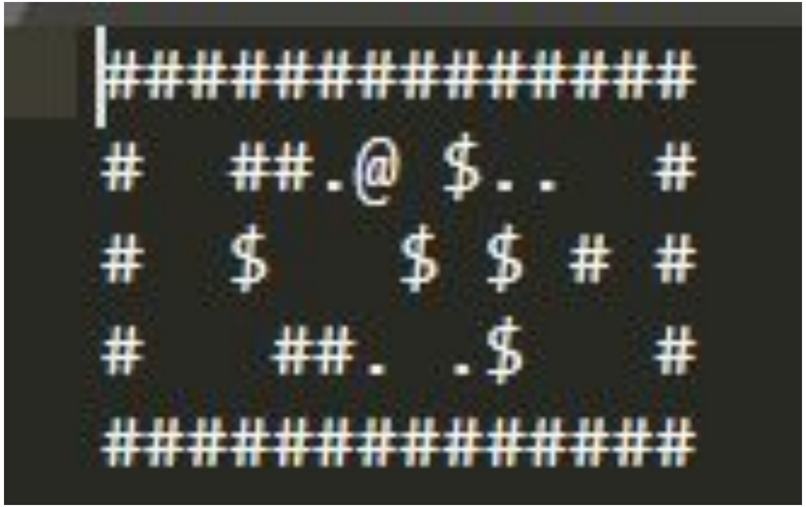
Sokoban Solvers

Grupo 6: Katan, Paganini

Resultados de los distintos métodos

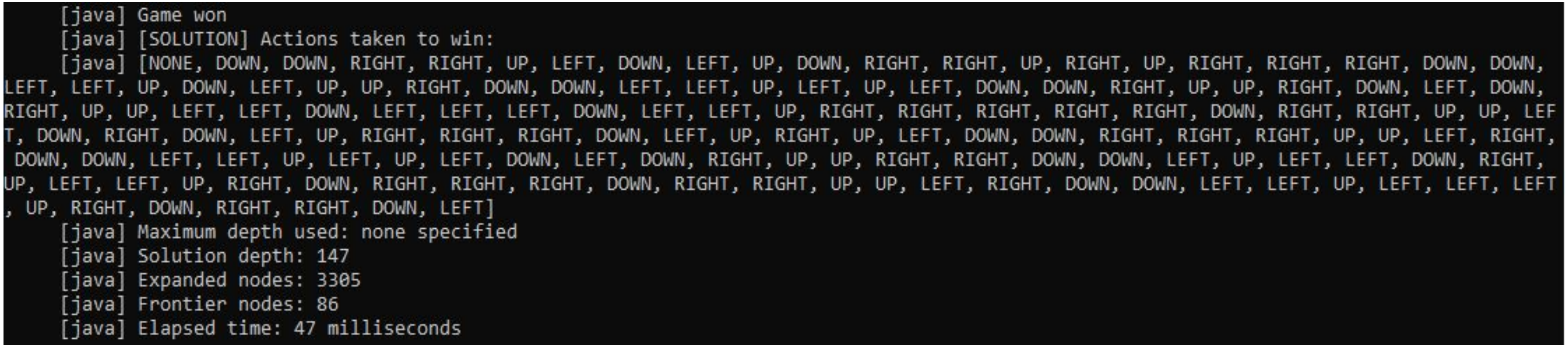
Usando el siguiente mapa:



```
##############
#  ##.@ $..  #
#  $   $ $ # #
#   ##. .$   #
##############
```

Métodos no informados - Resultados

- DFS:

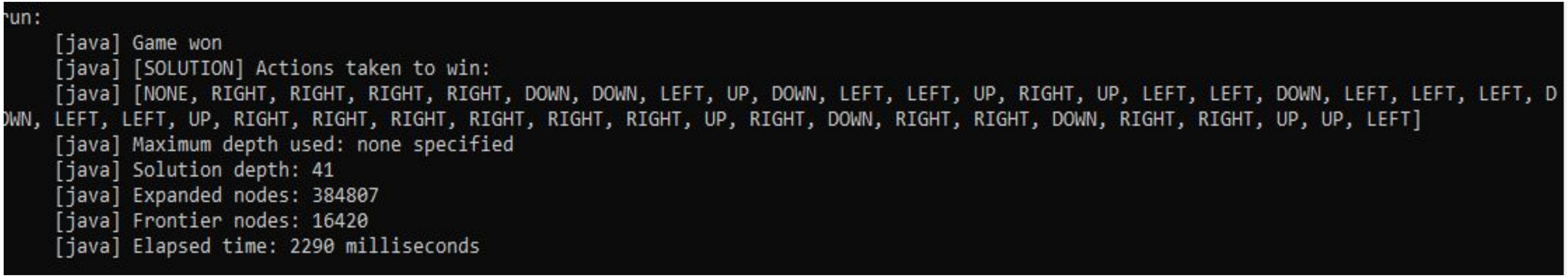


```
     [java] Game won
     [java] [SOLUTION] Actions taken to win:
     [java] [NONE, DOWN, DOWN, RIGHT, RIGHT, UP, LEFT, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, UP, RIGHT, UP, RIGHT, RIGHT, RIGHT, DOWN, DOWN,
LEFT, LEFT, UP, DOWN, LEFT, UP, UP, RIGHT, DOWN, DOWN, LEFT, LEFT, UP, LEFT, UP, LEFT, DOWN, DOWN, RIGHT, UP, UP, RIGHT, DOWN, LEFT, DOWN,
RIGHT, UP, UP, LEFT, LEFT, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, UP, UP, LEF
T, DOWN, RIGHT, DOWN, LEFT, UP, RIGHT, RIGHT, RIGHT, DOWN, LEFT, UP, RIGHT, UP, LEFT, DOWN, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT, RIGHT,
 DOWN, DOWN, LEFT, LEFT, UP, LEFT, UP, LEFT, DOWN, LEFT, DOWN, RIGHT, UP, UP, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, LEFT, LEFT, DOWN, RIGHT,
UP, LEFT, LEFT, UP, RIGHT, DOWN, RIGHT, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, UP, UP, LEFT, RIGHT, DOWN, DOWN, LEFT, LEFT, UP, LEFT, LEFT, LEFT
, UP, RIGHT, DOWN, RIGHT, RIGHT, DOWN, LEFT]
     [java] Maximum depth used: none specified
     [java] Solution depth: 147
     [java] Expanded nodes: 3305
     [java] Frontier nodes: 86
     [java] Elapsed time: 47 milliseconds
```

- Solución no óptima
- Más rápido que BFS
- Menos uso de memoria que BFS

Métodos no informados - Resultados

- BFS:

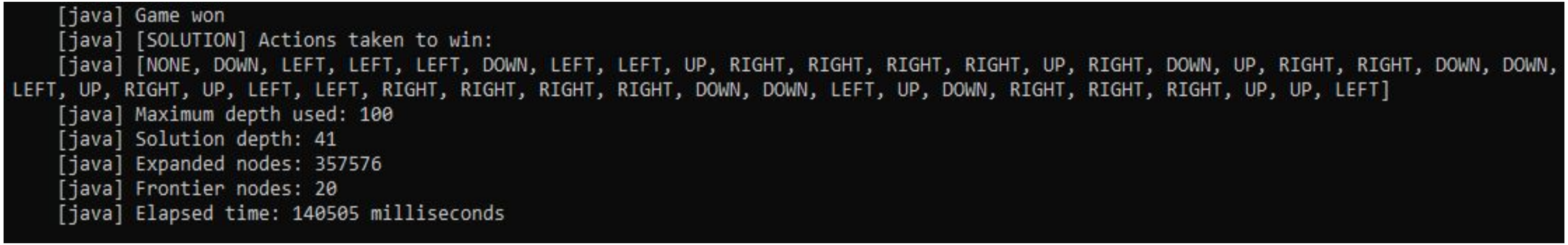


```
un:
    [java] Game won
    [java] [SOLUTION] Actions taken to win:
    [java] [NONE, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, LEFT, LEFT, UP, RIGHT, UP, LEFT, LEFT, DOWN, LEFT, LEFT, LEFT, D
WN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, DOWN, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, UP, UP, LEFT]
    [java] Maximum depth used: none specified
    [java] Solution depth: 41
    [java] Expanded nodes: 384807
    [java] Frontier nodes: 16420
    [java] Elapsed time: 2290 milliseconds
```

- Solución óptima
- Usa más memoria que DFS
- Es más lento que DFS

Métodos no informados - Resultados

- IDDFS:



```
    [java] Game won
    [java] [SOLUTION] Actions taken to win:
    [java] [NONE, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, DOWN, UP, RIGHT, RIGHT, DOWN, DOWN,
LEFT, UP, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT]
    [java] Maximum depth used: 100
    [java] Solution depth: 41
    [java] Expanded nodes: 357576
    [java] Frontier nodes: 20
    [java] Elapsed time: 140505 milliseconds
```

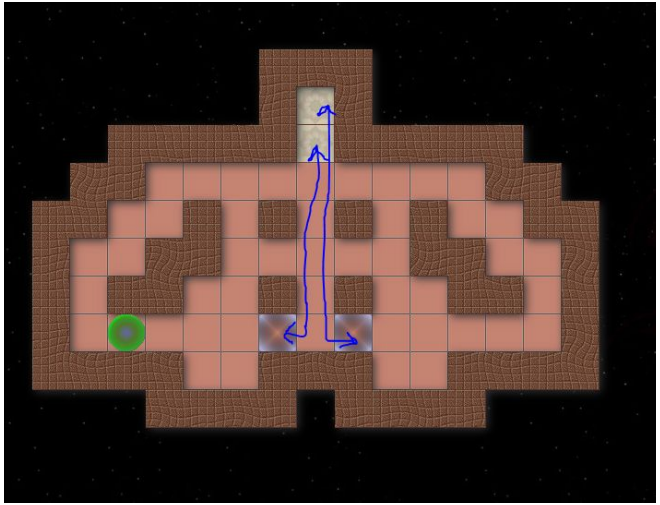
- Solución óptima
- Menor uso de memoria que BFS
- Tarda más tiempo que BFS

Métodos informados - Heurísticas

- Frozen Boxes
- Linear Conflicts
- Manhattan Distance

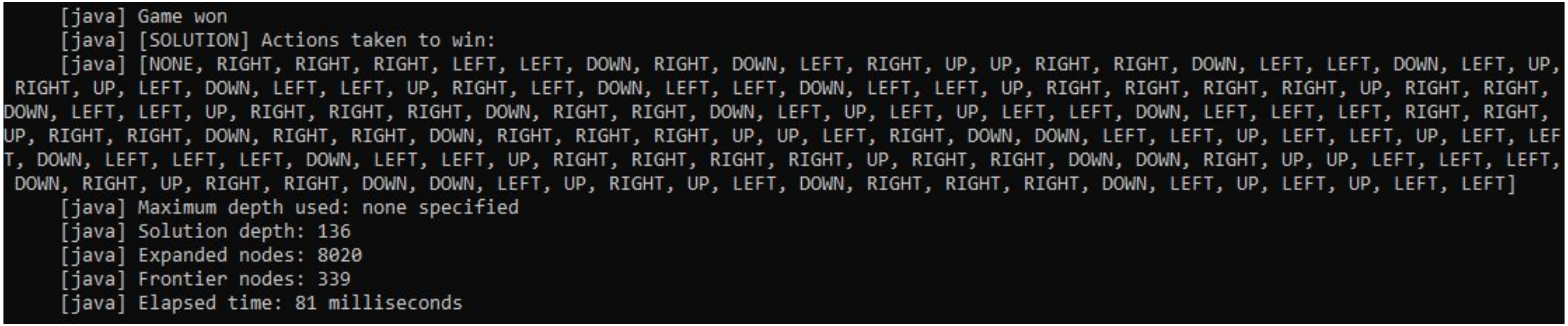
Métodos informados - Heurísticas y resultados

- Manhattan Distance: Suma de las distancias de manhattan mínimas de las cajas a los targets



Métodos informados - Heurísticas y resultados

- Manhattan Distance, resultados usando Global Greedy Search:

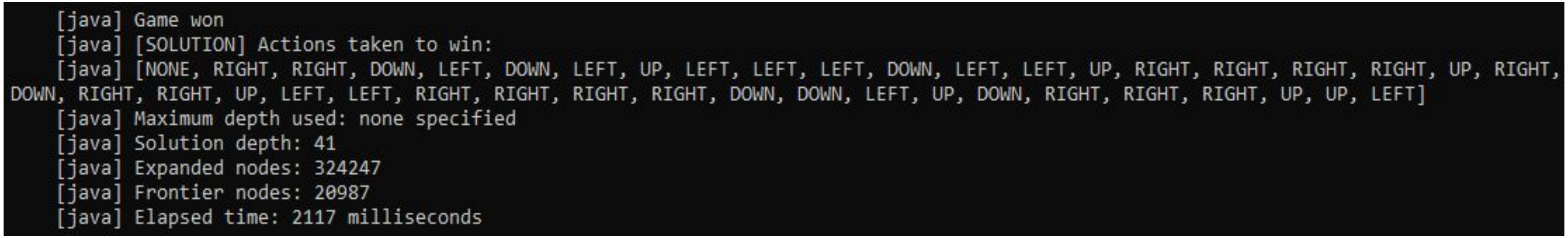


```
     [java] Game won
     [java] [SOLUTION] Actions taken to win:
     [java] [NONE, RIGHT, RIGHT, RIGHT, LEFT, LEFT, DOWN, RIGHT, DOWN, LEFT, RIGHT, UP, UP, RIGHT, RIGHT, DOWN, LEFT, LEFT, DOWN, LEFT, UP,
 RIGHT, UP, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, LEFT, DOWN, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, RIGHT,
DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, DOWN, LEFT, UP, LEFT, UP, LEFT, LEFT, DOWN, LEFT, LEFT, LEFT, RIGHT, RIGHT,
UP, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT, RIGHT, DOWN, DOWN, LEFT, LEFT, UP, LEFT, LEFT, UP, LEFT, LEF
T, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, RIGHT, DOWN, DOWN, RIGHT, UP, UP, LEFT, LEFT, LEFT,
 DOWN, RIGHT, UP, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, RIGHT, UP, LEFT, DOWN, RIGHT, RIGHT, RIGHT, DOWN, LEFT, UP, LEFT, UP, LEFT, LEFT]
     [java] Maximum depth used: none specified
     [java] Solution depth: 136
     [java] Expanded nodes: 8020
     [java] Frontier nodes: 339
     [java] Elapsed time: 81 milliseconds
```

- Solución no óptima, pero mejor que DFS
- Más lento que DFS
- Menos uso de memoria que A*

Métodos informados - Heurísticas y resultados

- Manhattan Distance, resultados usando A*:

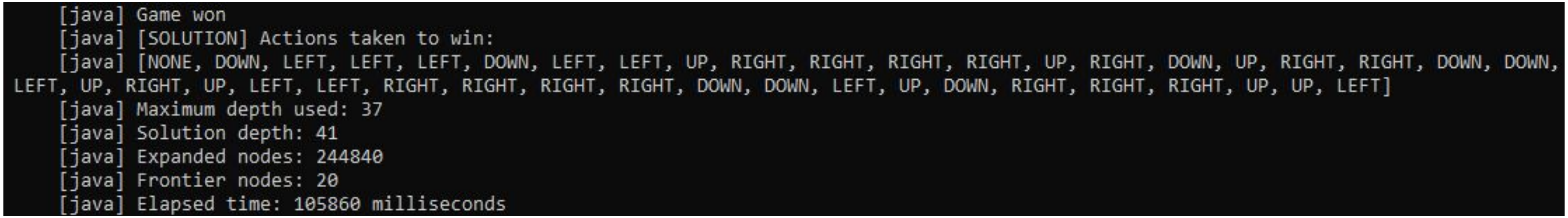


```
    [java] Game won
    [java] [SOLUTION] Actions taken to win:
    [java] [NONE, RIGHT, RIGHT, DOWN, LEFT, DOWN, LEFT, UP, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT,
DOWN, RIGHT, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT]
    [java] Maximum depth used: none specified
    [java] Solution depth: 41
    [java] Expanded nodes: 324247
    [java] Frontier nodes: 20987
    [java] Elapsed time: 2117 milliseconds
```

- Solución óptima
- Más lento que Global Greedy Search
- Usa más memoria que Global Greedy Search

Métodos informados - Heurísticas y resultados

- Manhattan Distance, resultados usando IDA*:

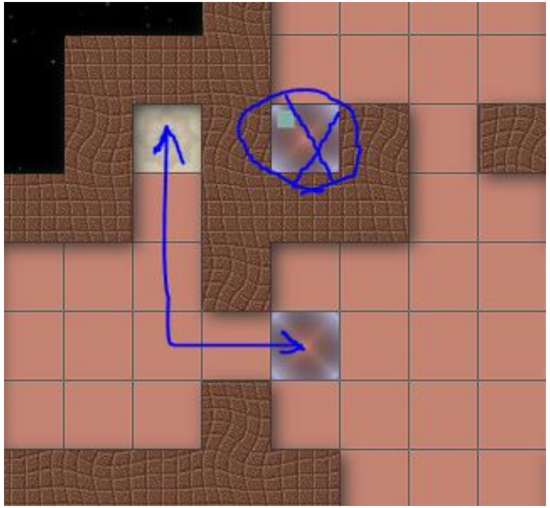


```
    [java] Game won
    [java] [SOLUTION] Actions taken to win:
    [java] [NONE, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, DOWN, UP, RIGHT, RIGHT, DOWN, DOWN,
LEFT, UP, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT]
    [java] Maximum depth used: 37
    [java] Solution depth: 41
    [java] Expanded nodes: 244840
    [java] Frontier nodes: 20
    [java] Elapsed time: 105860 milliseconds
```

- Resultado óptimo
- Menos uso de memoria que A*
- Tarda más tiempo que A *

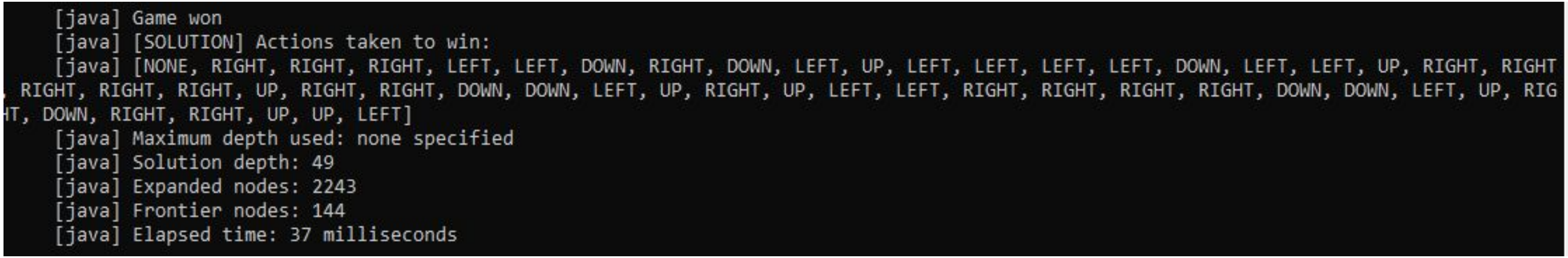
Métodos informados - Heurísticas y resultados

- Frozen boxes: Modificación sobre Manhattan Distance, descarta los targets que ya estén ocupados por cajas bloqueadas



Métodos informados - Heurísticas y resultados

- Frozen boxes, resultados usando Global Greedy Search:

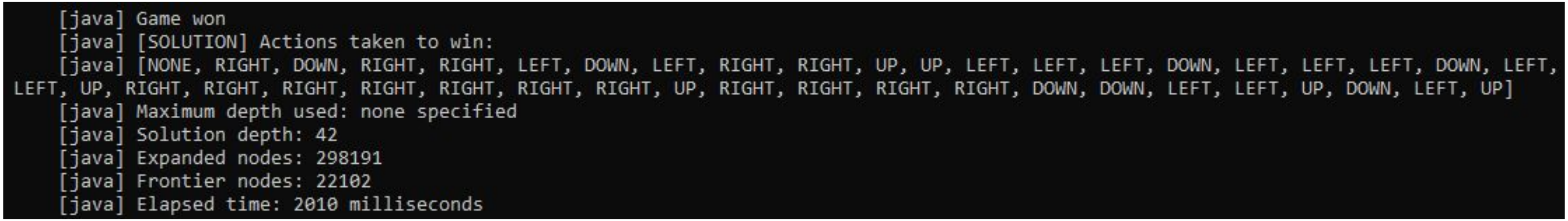


```
    [java] Game won
    [java] [SOLUTION] Actions taken to win:
    [java] [NONE, RIGHT, RIGHT, RIGHT, LEFT, LEFT, DOWN, RIGHT, DOWN, LEFT, UP, LEFT, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT
, RIGHT, RIGHT, RIGHT, UP, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, RIG
HT, DOWN, RIGHT, RIGHT, UP, UP, LEFT]
    [java] Maximum depth used: none specified
    [java] Solution depth: 49
    [java] Expanded nodes: 2243
    [java] Frontier nodes: 144
    [java] Elapsed time: 37 milliseconds
```

- Mejor solución que usando Global greedy search con la heurística Manhattan Distance, tanto en cantidad de movimientos, tiempo y memoria
- La heurística frozen boxes es útil en este mapa.

Métodos informados - Heurísticas y resultados

- Frozen boxes, resultados usando A*:

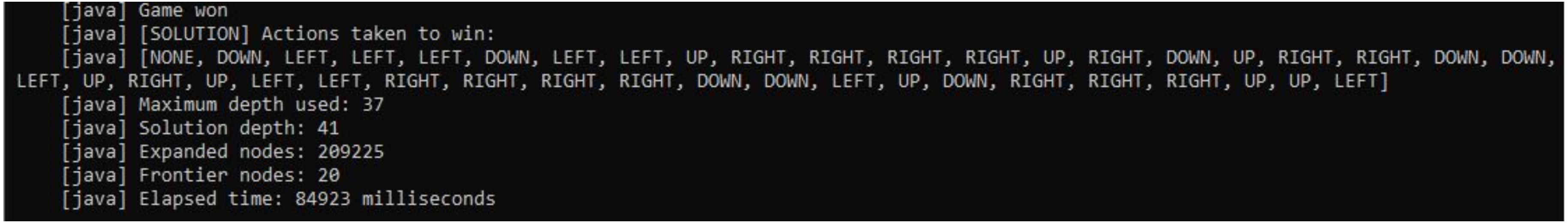


```
     [java] Game won
     [java] [SOLUTION] Actions taken to win:
     [java] [NONE, RIGHT, DOWN, RIGHT, RIGHT, LEFT, DOWN, LEFT, RIGHT, RIGHT, UP, UP, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT,
 LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, LEFT, UP, DOWN, LEFT, UP]
     [java] Maximum depth used: none specified
     [java] Solution depth: 42
     [java] Expanded nodes: 298191
     [java] Frontier nodes: 22102
     [java] Elapsed time: 2010 milliseconds
```

- Menor uso de memoria y menor tiempo que usando A* con Manhattan Distance

Métodos informados - Heurísticas y resultados

- Frozen boxes, resultados usando IDA*:

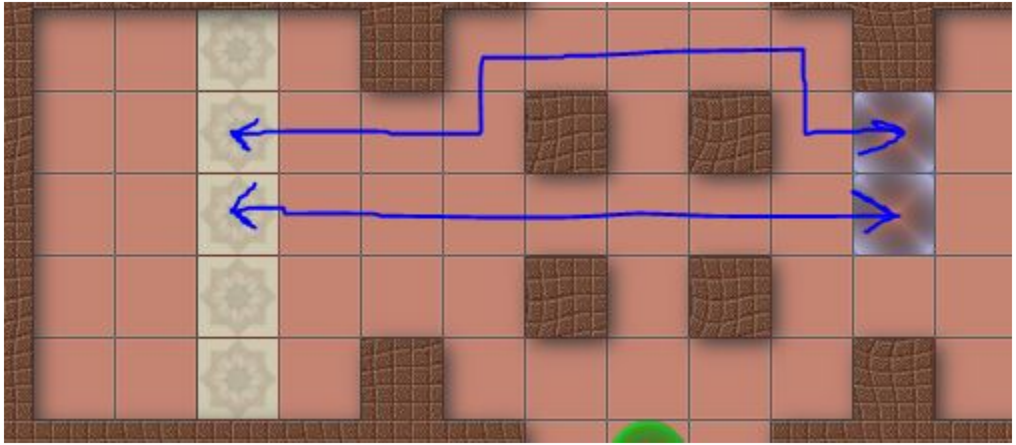


```
     [java] Game won
     [java] [SOLUTION] Actions taken to win:
     [java] [NONE, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, DOWN, UP, RIGHT, RIGHT, DOWN, DOWN,
LEFT, UP, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT]
     [java] Maximum depth used: 37
     [java] Solution depth: 41
     [java] Expanded nodes: 209225
     [java] Frontier nodes: 20
     [java] Elapsed time: 84923 milliseconds
```

- Más rápido y menos uso de memoria que usando IDA* con Manhattan Distance

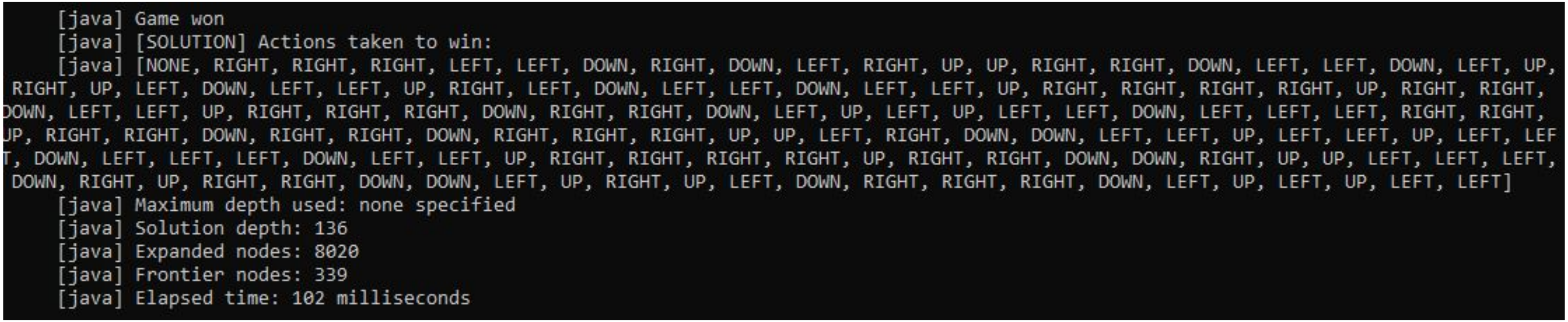
Métodos informados - Heurísticas y resultados

- Linear conflicts: Modificación sobre Manhattan Distance, teniendo en cuenta obstáculos sobre un camino lineal hacia el target



Métodos informados - Heurísticas y resultados

- Linear conflicts, resultados usando Global Greedy Search:

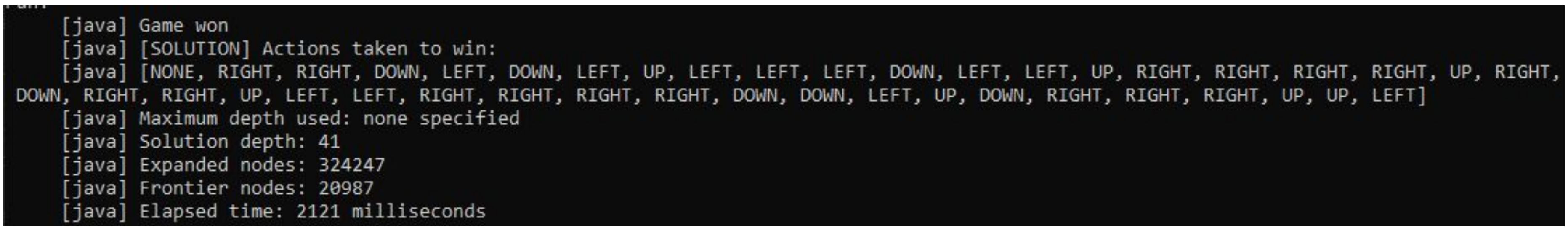


```
     [java] Game won
     [java] [SOLUTION] Actions taken to win:
     [java] [NONE, RIGHT, RIGHT, RIGHT, LEFT, LEFT, DOWN, RIGHT, DOWN, LEFT, RIGHT, UP, UP, RIGHT, RIGHT, DOWN, LEFT, LEFT, DOWN, LEFT, UP,
 RIGHT, UP, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, LEFT, DOWN, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, RIGHT,
DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, DOWN, LEFT, UP, LEFT, UP, LEFT, LEFT, DOWN, LEFT, LEFT, LEFT, RIGHT, RIGHT,
UP, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT, RIGHT, DOWN, DOWN, LEFT, LEFT, UP, LEFT, LEFT, UP, LEFT, LEF
T, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, RIGHT, DOWN, DOWN, RIGHT, UP, UP, LEFT, LEFT, LEFT,
 DOWN, RIGHT, UP, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, RIGHT, UP, LEFT, DOWN, RIGHT, RIGHT, RIGHT, DOWN, LEFT, UP, LEFT, UP, LEFT, LEFT]
     [java] Maximum depth used: none specified
     [java] Solution depth: 136
     [java] Expanded nodes: 8020
     [java] Frontier nodes: 339
     [java] Elapsed time: 102 milliseconds
```

- Mismo resultado que usando solo Manhattan Distance
- Al no haber muchos caminos directos entre caja y target, esta heurística no mejora sustancialmente la búsqueda

Métodos informados - Heurísticas y resultados

- Linear conflicts, resultados usando A*:

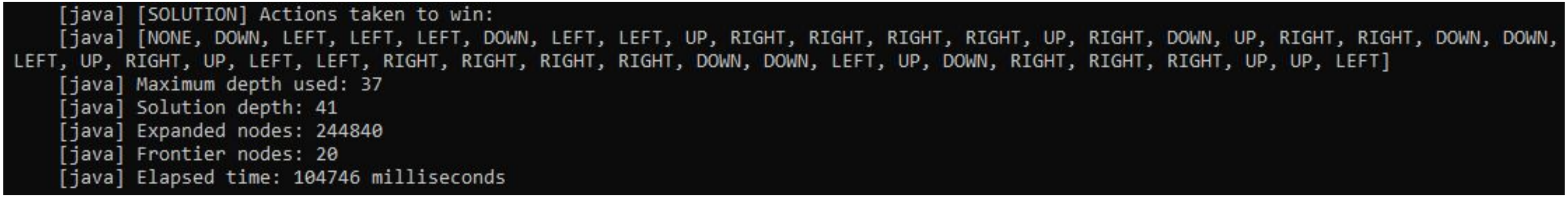


```
     [java] Game won
     [java] [SOLUTION] Actions taken to win:
     [java] [NONE, RIGHT, RIGHT, DOWN, LEFT, DOWN, LEFT, UP, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT,
 DOWN, RIGHT, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT]
     [java] Maximum depth used: none specified
     [java] Solution depth: 41
     [java] Expanded nodes: 324247
     [java] Frontier nodes: 20987
     [java] Elapsed time: 2121 milliseconds
```

- Nuevamente, vemos que esta heurística no ofrece una mejora sustancial para este mapa en particular.

Métodos informados - Heurísticas y resultados

- Linear conflicts, resultados usando IDA*:



```
    [java] [SOLUTION] Actions taken to win:
    [java] [NONE, DOWN, LEFT, LEFT, LEFT, DOWN, LEFT, LEFT, UP, RIGHT, RIGHT, RIGHT, RIGHT, UP, RIGHT, DOWN, UP, RIGHT, RIGHT, DOWN, DOWN,
LEFT, UP, RIGHT, UP, LEFT, LEFT, RIGHT, RIGHT, RIGHT, RIGHT, DOWN, DOWN, LEFT, UP, DOWN, RIGHT, RIGHT, RIGHT, UP, UP, LEFT]
    [java] Maximum depth used: 37
    [java] Solution depth: 41
    [java] Expanded nodes: 244840
    [java] Frontier nodes: 20
    [java] Elapsed time: 104746 milliseconds
```

- Nuevamente, vemos que esta heurística no ofrece una mejora sustancial para este mapa en particular.