



Tecnológico de Monterrey

Sistemas Inteligentes

Documento - Parcial 1

Román Andrés Aguilar Hernández
Ricardo Huerta Dórame

A01324582
A00397978

Problema

El problema a resolver en esta ocasión es el rompecabezas numérico de 3*3. Este consiste en tener una matriz de 3*3 la cual tiene 8 números y un espacio en blanco. El objetivo de este problema es llegar a una matriz objetivo solamente cambiando de lugar los números que son vecinos de la casilla con valor en blanco. Por ejemplo, teniendo la matriz

1,2,3
4,5,6
7,8,0

Un movimiento válido sería mover el 8 a la derecha y el espacio en blanco a la izquierda de la siguiente manera

1,2,3
4,5,6
7,0,8

Para motivos de este examen no solo tendremos matrices de 3*3, también tendremos de 4*4 y 5*5.

Solución

Para la solución de este problema desarrollamos dos funciones heurísticas para desarrollar nuestra función f.

$$f = h1(\text{someList}) + h2(\text{someList})$$

Para f decidimos calcular el número de posiciones incorrectas que hay en el tablero. De igual manera obtenemos el número de pasos directos que se tienen que hacer para alcanzar la posición final de nuestra tabla objetivo. Esta función es buena porque estamos cubriendo los casos que consideramos que son los más importantes a la hora de razonar qué hay que hacer para llegar a la tabla objetivo. Por ejemplo:

Teniendo la tabla

1,2,3
4,5,6
7,8,0

Y como tabla objetivo

1,2,3
8,0,4
7,6,5

Podemos determinar que el número de posiciones incorrectas es 5 ya que los números 1,2,3,7 ya se encuentran en la casilla que pertenecen. A su vez el número de pasos directos que se requieren es de 10.

Pseudocódigo

Node definition:

- Index of father
- Movement
- Heuristic value
- G value
- Board
- Id
- Father node

read file and create two matrixes (initialMatrix, goalMatrix)

declare open

declare closed

print initial matrix

create first node and append it to open

open := [start]

closed := []

while open is different to []

 remove state from open, call it x;

 if x = goal

 print path to X

 print matrix

 exit

 else

 generate children of x

 set childInOpen and childInClosed to false

 foreach child of x

 foreach node in open

 if child is on open

 childInOpen = True

 if child path is shorter than the one in open

 give the state on open the shorter path

 foreach node in open

 if child is on closed

 childInClosed = True

 if child path is shorter than the one in closed

 kill children from this parent to generate the again

 remove state from closed

 assign heuristic value to child

 add child to open

 if child is not on open or closed

 assign heuristic value to child

 add child to open

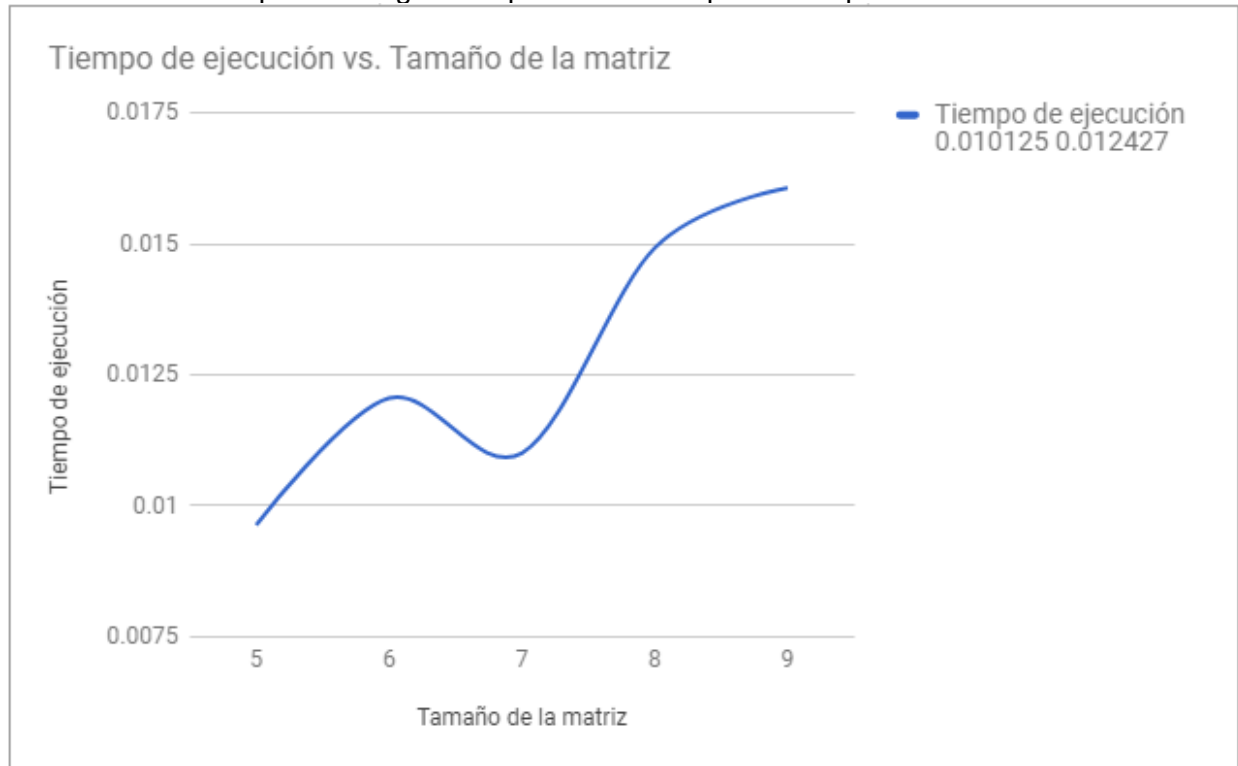
 set childInOpen and childInClosed to false

put x on closed

sort open

Capturas de pantalla / Resultados

Con base a la tabla de la izquierda nos podemos dar cuenta de que en matrices donde n es un valor par el tiempo crece comparado a cuando n es un valor non (el tiempo decrece). También nos damos cuenta que esta regla se repite en valores pares comparándolos con $n-1$



| Tamaño de la matriz | Tiempo de ejecución |
|---------------------|---------------------|
| 3 | 0.010125 |
| 4 | 0.012427 |
| 5 | 0.009635 |
| 6 | 0.012055 |
| 7 | 0.011005 |
| 8 | 0.014925 |
| 9 | 0.01607 |

```

toroidalhawk@RHD-Laptop:/mnt/c/Users/cafu9/Documents/9noSemestre/Sistemas/1erParcial/BFS-puzzle-solver$ python parcial1.py texto.txt
ARREGLO INICIAL
  4  1  0
  3  5  6
  7  8  2
RESPUESTA:
  1  2  3
  4  5  6
  7  8  0
La salida es: L,D,L,U,R,R,D,D,L,U,R,U,L,D,D,R,U,L,U,R,D,L,L,U,R,R,D,D,L,U,L,D,R,R,U,U,L,D,R,D,L,U,L,D,R,U,U,L,D,D,R,U,L,U,R,D,D,R
toroidalhawk@RHD-Laptop:/mnt/c/Users/cafu9/Documents/9noSemestre/Sistemas/1erParcial/BFS-puzzle-solver$

```

3
 4,1,0
 3,5,6
 7,8,2
 1,2,3
 4,5,6
 7,8,0

```

toroidalhawk@RHD-Laptop:/mnt/c/Users/cafu9/Documents/9noSemestre/Sistemas/1erParcial/BFS-puzzle-solver$ python parcial1.py texto.txt
ARREGLO INICIAL
  1  2  3  4
  0  9  7  8
  6  5 11 12
 13 14 15 16
RESPUESTA:
  1  2  3  4
  5  6  7  8
  9  0 11 12
 13 14 15 16
La salida es: D,R,U,L,D,R
toroidalhawk@RHD-Laptop:/mnt/c/Users/cafu9/Documents/9noSemestre/Sistemas/1erParcial/BFS-puzzle-solver$

```

4
 1,2,3,4
 0,9,7,8
 6,5,11,12
 13,14,15,16
 1,2,3,4
 5,6,7,8
 9,0,11,12
 13,14,15,16

```

toroidalhawk@RHD-Laptop:/mnt/c/Users/cafu9/Documents/9noSemestre/Sistemas/1erParcial/BFS-puzzle-solver$ python parcial1.py texto.txt
ARREGLO INICIAL
  1  2  3  4  5
  6  7  8  9 10
 11 12 13 14 15
 16 17 18 19 20
 21 22 23 24  0
RESPUESTA:
  0  2  3  4  5
  1  7  8  9 10
  6 12 13 14 15
 11 17 18 19 20
 16 21 22 23 24
La salida es: L,L,L,L,U,U,U,U
toroidalhawk@RHD-Laptop:/mnt/c/Users/cafu9/Documents/9noSemestre/Sistemas/1erParcial/BFS-puzzle-solver$

```

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

Otros tableros

6
 1,2,3,4,5,25
 6,7,8,9,10,26
 11,12,13,14,15,27
 16,17,18,19,20,28
 21,22,23,24,0,29
 30,31,32,33,34,35
 0,2,3,4,5,25
 1,7,8,9,10,26
 6,12,13,14,15,27
 11,17,18,19,20,28
 16,21,22,23,24,29
 30,31,32,33,34,35

7

1,2,3,4,5,25,48
6,7,8,9,10,26,47
11,12,13,14,15,27,46
16,17,18,19,20,28,45
21,22,23,24,0,29,44
30,31,32,33,34,35,43
36,37,38,39,40,41,42
0,2,3,4,5,25,48
1,7,8,9,10,26,47
6,12,13,14,15,27,46
11,17,18,19,20,28,45
16,21,22,23,24,29,44
30,31,32,33,34,35,43
36,37,38,39,40,41,42

9

1,2,3,4,5,25,48,49,80
6,7,8,9,10,26,47,50,79
11,12,13,14,15,27,46,51,78
16,17,18,19,20,28,45,52,77
21,22,23,24,0,29,44,53,76
30,31,32,33,34,35,43,54,75
36,37,38,39,40,41,42,55,74
56,57,58,59,60,61,62,63,73
64,65,66,67,68,69,70,71,72
0,2,3,4,5,25,48,49,80
1,7,8,9,10,26,47,50,79
6,12,13,14,15,27,46,51,78
11,17,18,19,20,28,45,52,77
16,21,22,23,24,29,44,53,76
30,31,32,33,34,35,43,54,75
36,37,38,39,40,41,42,55,74
56,57,58,59,60,61,62,63,73
64,65,66,67,68,69,70,72