

Inteligencia Artificial

Trabajo Práctico 1

8 Puzzle

A: Desordenar el 8 puzzle realizando al menos 50 movimientos aleatorios, a partir de la posición "objetivo".

B: Desordenar el puzzle por el método anterior. Activar búsqueda aleatoria y verificar en cuántos movimientos se encuentra la solución (puede no encontrarla). Correr la búsqueda 10 veces e informar el promedio de movimientos empleados.

C: Desordenar el puzzle. Activar búsqueda en anchura. Correrla 10 veces e informar en qué promedio de movimientos encuentra el resultado.

A continuación se mostrarán los resultados de la ejecución del programa incluyendo hasta el ítem C. Se podrá observar cada iteración y finalmente los tiempos y cantidades de movimientos obtenidos. ***Resultados finales: Página 15**

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

1 2 3

4 6

7 5 8

Búsqueda aleatoria:

[1, 2, 3]

[4, 6, 0]

[7, 5, 8]

Solución random encontrada en 693388 movimientos!

Tiempo empleado para encontrar la solución random: 0.993799 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

El camino de la solución encontrada tiene 3 pasos

El nivel actual del árbol (3) tiene 9 ramas

Solución en anchura encontrada en 17 movimientos

Tiempo para encontrar solución por anchura: 0.000497 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

4 1 2

7 5

8 6 3

Búsqueda aleatoria:

[4, 1, 2]

[7, 5, 0]

[8, 6, 3]

Solución random encontrada en 4742 movimientos!

Tiempo empleado para encontrar la solución random: 0.007039 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

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

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

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

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

El camino de la solución encontrada tiene 9 pasos

El nivel actual del árbol (9) tiene 152 ramas

Solución en anchura encontrada en 390 movimientos

Tiempo para encontrar solución por anchura: 0.024641 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

2 3

1 4 6

7 5 8

Búsqueda aleatoria:

[2, 0, 3]

[1, 4, 6]

[7, 5, 8]

Solución random encontrada en 14 movimientos!

Tiempo empleado para encontrar la solución random: 0.000024 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

El camino de la solución encontrada tiene 5 pasos

El nivel actual del árbol (5) tiene 23 ramas

Solución en anchura encontrada en 51 movimientos

Tiempo para encontrar solución por anchura: 0.001622 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

1 3 5

4 2

7 8 6

Búsqueda aleatoria:

[1, 3, 5]

[4, 2, 0]

[7, 8, 6]

Solución random encontrada en 4139544 movimientos!

Tiempo empleado para encontrar la solución random: 6.085345 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

El camino de la solución encontrada tiene 5 pasos

El nivel actual del árbol (5) tiene 17 ramas

Solución en anchura encontrada en 39 movimientos

Tiempo para encontrar solución por anchura: 0.001384 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

4 1 2

5 8

7 6 3

Búsqueda aleatoria:

[4, 1, 2]

[5, 8, 0]

[7, 6, 3]

Solución random encontrada en 46838 movimientos!

Tiempo empleado para encontrar la solución random: 0.069346 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

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

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

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

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

El camino de la solución encontrada tiene 9 pasos

El nivel actual del árbol (9) tiene 141 ramas

Solución en anchura encontrada en 366 movimientos

Tiempo para encontrar solución por anchura: 0.023295 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

1 2 3

4 5

7 8 6

Búsqueda aleatoria:

[1, 2, 3]

[4, 5, 0]

[7, 8, 6]

Solución random encontrada en 1485152 movimientos!

Tiempo empleado para encontrar la solución random: 2.195272 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

El camino de la solución encontrada tiene 1 pasos

El nivel actual del árbol (1) tiene 2 ramas

Solución en anchura encontrada en 2 movimientos

Tiempo para encontrar solución por anchura: 0.000045 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

1 6 2

5 7 3

4 8

Búsqueda aleatoria:

[1, 6, 2]

[5, 7, 3]

[4, 0, 8]

Solución random encontrada en 386872 movimientos!

Tiempo empleado para encontrar la solución random: 0.571565 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

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

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

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

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

El camino de la solución encontrada tiene 9 pasos

El nivel actual del árbol (9) tiene 124 ramas

Solución en anchura encontrada en 325 movimientos

Tiempo para encontrar solución por anchura: 0.021295 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

1 3 5

4 2 6

7 8

Búsqueda aleatoria:

[1, 3, 5]

[4, 2, 6]

[7, 0, 8]

Solución random encontrada en 198520 movimientos!

Tiempo empleado para encontrar la solución random: 0.293412 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

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

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

El camino de la solución encontrada tiene 7 pasos

El nivel actual del árbol (7) tiene 74 ramas

Solución en anchura encontrada en 171 movimientos

Tiempo para encontrar solución por anchura: 0.007617 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

4 1

5 7 3

2 8 6

Búsqueda aleatoria:

[4, 0, 1]

[5, 7, 3]

[2, 8, 6]

Solución random encontrada en 243872 movimientos!

Tiempo empleado para encontrar la solución random: 0.359128 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

El camino de la solución encontrada tiene 15 pasos

El nivel actual del árbol (15) tiene 3000 ramas

Solución en anchura encontrada en 7822 movimientos

Tiempo para encontrar solución por anchura: 2.890050 segundos

Tablero inicial (condición de victoria):

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

1 3 5

8 2

4 7 6

Búsqueda aleatoria:

[1, 3, 5]

[8, 2, 0]

[4, 7, 6]

Solución random encontrada en 441752 movimientos!

Tiempo empleado para encontrar la solución random: 0.651765 segundos

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Búsqueda por anchura, buscando caminos...

Victoria

Camino de la solución encontrada:

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

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

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

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

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

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

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

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

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

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

El camino de la solución encontrada tiene 9 pasos

El nivel actual del árbol (9) tiene 119 ramas

Solución en anchura encontrada en 317 movimientos

Tiempo para encontrar solución por anchura: 0.021102 segundos

Se realiza 10 veces el siguiente proceso: desordenar el tablero inicial con 30 movimientos aleatorios y se realiza la búsqueda random y luego por anchura

Una vez encontrado los resultados se calcula el promedio de tiempo para la búsqueda random y el promedio de movimientos realizados

De igual manera se calcula el promedio de tiempo para la búsqueda por anchura y el promedio de movimientos realizados

Búsqueda random: 764069.4 movimientos promedio y 1.122670 segundos tiempo promedio de búsqueda.

Búsqueda por anchura: 950.0 movimientos promedio y 0.299155 segundos tiempo promedio de búsqueda.

D: Desordenar el puzzle. Activar búsqueda bidireccional. Correrla 10 veces e informar en qué promedio de movimientos encuentra el resultado.