

Sudoku Samurai: Backtracking y A*

1 Objetivo

Con esta actividad vas a conseguir implementar la estrategia de búsqueda heurística **A*** para la resolución del problema del sudoku, en su variante Samurai.

2 Descripción de la actividad

Puede observarse un ejemplo de la aplicación requerida en el siguiente link:

El objetivo del juego consiste en colocar los números de 1 a 9 en cinco tablero de 9×9 casillas interconectados sin que se repitan números en filas, columnas ni regiones. En esta variante hay 5 sudokus interconectados que hay que resolver en conjunto y que tiene una solución única.

Se presenta un tablero con cinco sudokus de tamaño 9×9 divididos en regiones de igual tamaño. El juego consiste en colocar los números de 1 a 9 de manera que no se repitan en ninguna fila ni columna ni región.

En esta variante hay 5 sudokus interconectados que no pueden solucionarse de forma individual pero sí tienen una solución conjunta. Los sudokus están conectados en forma de *X*

Author: Otto Janko

La solución para el tablero presentado anteriormente es el siguiente:

6	3	1	4	7	5	9	8	2
---	---	---	---	---	---	---	---	---

3	2	6	7	4	8	9	5	1
---	---	---	---	---	---	---	---	---

9	4	3	7	2	6	1	5	0
---	---	---	---	---	---	---	---	---

8	4	3	2	1	7	6	5	9
---	---	---	---	---	---	---	---	---

En el caso de **A***, en cada iteración del algoritmo se deberá indicar claramente el nodo que ha sido expan-

Se deberán crear tantas clases o estructuras de datos como sean necesarias para representar el espacio de

estados y los nodos de exploración del árbol.

3 Interfaz gráfica

Se debe presentar el tablero en una página HTML desde donde se importen los archivos Javascript necesarios, la interfaz puede ser sencilla pero debe permitir **avanzar por pasos o de manera continua**.

El programa debe ser capaz de resolver **problemas generados de forma aleatoria** o alguno introducido por el usuario.

Puede utilizar librerías externas para la representación visual, pero la implementación de los algoritmos debe ser un trabajo totalmente original de los estudiantes.

Al alcanzar el resultado, se debe indicar la secuencia de acciones a realizar para alcanzar el objetivo utilizando una notación sencilla.

El programa desarrollado debe ser un trabajo original del estudiante.

El código debe venir debidamente comentado, se debe utilizar **jsDoc** para cada función o clase utilizada.

4 Checklist

- Interfaz
 - Tablero de tamaño dinámico
 - Secuencia paso a paso de la búsqueda de solución (en el tablero y terminal)
 - Secuencia automática con la solución.
 - Generar tablero de forma aleatoria
 - Permitir al usuario introducir un tablero al inicio.
- Funcionalidades
 - Resolución por Backtracking
 - Resolución por A*

5 Nota importante

Existen mucho código en internet con soluciones al problema, por lo que durante la revisión se harán preguntas para comprobar el conocimiento del código por cada uno de los estudiantes. Aunque no es requisito, si los estudiantes utilizan Git para el trabajo colaborativo del código, se puede utilizar para comprobar el trabajo de cada uno.

Es probable que la ejecución de backtracking provoque que la computadora se quede sin recursos, por lo que es suficiente demostrar su correcto funcionamiento más que llegar a una solución de cada tablero.

6 Entregable

El proyecto deberá entregarse en el TecDigital, en un archivo comprimido que incluya los archivos necesarios para la ejecución. Se debe precisar en un archivo Readme.md cualquier iteración necesaria del usuario para la ejecución del programa.

Si la entrega se realiza después de la hora de entrega, se le penalizará con 5 puntos porcentuales que se acumulan cada 24 horas. Por ejemplo si entrega a las 10:05pm su evaluación tendrá una nota base de 95%, si entrega después de las 10:05 p.m. del siguiente día, su nota base será 90%, y así sucesivamente.

Se puede realizar en grupos de 4.

7 Evaluación

Actividad	Descripción	Peso%
Criterio 1	Implementación del algoritmo Backtracking	30%
	Implementación del algoritmo A*:	
Criterio 2	el cálculo de la función de evaluación de los nodos $f(n)$ es correcta y el contenido de la lista abierta y de la lista cerrada para cada iteración es correcta	40%
Criterio 3	La interfaz gráfica cumple con lo solicitado	20%
Criterio 3	El resultado obtenido, es decir la secuencia de acciones a realizar para alcanzar el objetivo, es correcto	10%
		100 %