



Práctica 1 :

Resolución de problemas y búsqueda.

Búsqueda no informada

1. Objetivo de la práctica

Familiarizarse con el código en java para resolver problemas de búsqueda. El código lo puedes encontrar en <http://code.google.com/p/aima-java/>. Descarga la última versión oficial [aima-java-1.7.0-Chp4-Complete.zip](#)

En esta práctica trataremos de aplicar diferentes técnicas de búsqueda para resolver el 8-puzzle, un juego de mesa para un único jugador que consiste en 8 piezas cuadradas numeradas del 1 al 8 y un espacio vacío en un tablero de 3 x 3. El objetivo del juego es, partiendo de un estado representado por una disposición determinada de las piezas (estado inicial I) realizar movimientos permitidos para alcanzar una configuración deseada (meta O) en el menor número de movimientos posible.

Se pretende (i) utilizar distintos algoritmos de búsqueda (ciega e informada), (ii) evaluar cada uno de ellos y (iii) extraer algunas conclusiones.

8-puzzle

La descripción de un estado especifica la localización de cada una de las 8 fichas y el espacio en blanco, en cada uno de los nueve cuadrados.

7	2	4
5		6
8	3	1

1	2	3
4	5	6
7	8	

Figura 1: Ejemplos de estado inicial y estado final del 8-puzzle

Un **estado inicial** es cualquier configuración de las fichas de partida y un **estado meta**, una configuración determinada que hay que alcanzar (ver Figura 1). Para representar el problema podemos utilizar un array con los números {0,1,2,3,4,5,6,7,8} donde el cero simboliza la casilla vacía.

2. Tareas

1. A partir de un tablero inicial generado aleatoriamente, aplicar las estrategias vistas en teoría relacionadas con métodos de búsqueda ciega (recorrido en anchura, profundidad en árbol y en grafo, búsqueda en profundidad recursiva con límite (DLS), y la búsqueda iterativa en profundidad IDLS) .
2. En las demos sólo se muestra la lista de acciones cuando se encuentra la solución. Mostrar los estados por los que se pasa al aplicar las soluciones. Define un método `executeActions`, al que pasamos como argumentos la lista de acciones solución y el problema.
3. Visualiza el número de nodos generados además de los expandidos y el tiempo de ejecución en milisegundos.
4. Aplicar las nociones y métodos aprendidos en los apartados anteriores para resolver el “problema de las fichas”, una simplificación unidimensional del 8-puzzle, donde partimos de una situación inicial,

```
+---+---+---+---+---+---+---+
| B | B | B |   | V | V | V |
+---+---+---+---+---+---+---+
```

queremos alcanzar una situación final:

```
+---+---+---+---+---+---+---+
| V | V | V |   | B | B | B |
+---+---+---+---+---+---+---+
```

y los movimientos permitidos consisten en desplazar una ficha al hueco, saltando como máximo, sobre otras dos.

Se deben entregar en un zip con las clases escritas o modificadas y una memoria escrita con las tareas realizadas comentando los resultados obtenidos al utilizar distintos algoritmos.