

Sistemas de Inteligencia Artificial

Métodos de Búsqueda No Informados

Informe Preliminar 1

Objetivo

Se debe crear un Sistema de Producción que será usado para resolver el problema asignado a cada grupo.

Se les entregará un motor de inferencia reducido programado en Java. Cada grupo puede decidir entre utilizar este motor y hacerle las modificaciones que sean necesarias para completar el trabajo o realizar dicho motor en el lenguaje que deseen.

Descripción del trabajo

Se deberá realizar lo siguiente:

- Implementación de las estrategias de búsqueda no informadas: *depth first*, *breadth first* y *profundización iterativa*.
- **Heurísticas.** Presentación de al menos dos (2) heurísticas. La/las heurísticas deben ser no triviales. Enumerar las diferencias entre ellas. No es necesaria la implementación de las heurísticas, pero sí la explicación de cada una y sus diferencias.
- **Función de costo.** Presentación de la/las funciones de costo. Si hay más de una, enumerar las diferencias entre ellas. No es necesaria la implementación de las funciones de costo pero sí la explicación de cada una y sus diferencias.
- De cada corrida se deberá analizar al menos: la profundidad de la solución, cantidad total de estados generados, número de nodos frontera, número de nodos expandidos y tiempo de procesamiento.

El informe deberá describir el trabajo realizado, el análisis de los resultados obtenidos, las heurísticas, las funciones de costo y las conclusiones. La longitud del mismo no puede ser mayor a 2 carillas (se penalizará con 1 punto por cada carilla excedida). Las imágenes explicativas y tablas pueden estar en una sección anexa y no cuentan en el total de hojas.

No está permitido el uso de librerías o *toolkits*.

Forma de entrega

La entrega estará dividida en dos partes:

1. *Parte impresa.* Un informe impreso detallando el desarrollo. Deberá describir brevemente la implementación realizada además de cómo se resolvió el problema asignado con las justificaciones necesarias. **No** deberá exceder las dos carillas.

NOTA: Se descontará 1 punto por cada carilla adicional que se entregue.

El informe debe estar debidamente organizado: introducción, desarrollo, resultados y conclusiones.

2. *Parte digital.* (una revisión SVN con el trabajo completo):

- Documentos mencionados en la sección anterior.
- Códigos fuente del trabajo.
- Binarios ejecutables (incluyendo librerías, plugins y otros elementos necesarios para su ejecución).
- Un README con una descripción del procedimiento necesario para ejecutar el programa. Si al seguir los pasos especificados en el README el trabajo no funciona, se considera reprobado.

Fecha de entrega

Jueves 15 de Marzo a las 19:05.

NOTA: no se aceptarán trabajos luego de este horario, así como tampoco entregas parciales del mismo.

Problemas a resolver

Como temática para este trabajo, se decidió que cada grupo realice la resolución de alguno de los siguientes juegos:

1. **CalcuDoku:** Cada *puzzle* consiste de una grilla bloques. El objetivo es completar todos los espacios vacíos para que los números del 1 a N (siendo N la cantidad de filas o columnas en la grilla), aparezcan exactamente una vez en cada fila y columna. Además, los números en cada bloque deberán producir el resultado que se muestra en el bloque, aplicando la operación matemática asignada a la grilla.
URL: <http://www.conceptispuzzles.com/index.aspx?uri=puzzle/calculodoku>
2. **Eternity II:** Eternity II es un rompecabezas donde hay que acomodar 256 fichas cuadradas en una grilla de 16 por 16, teniendo en cuenta que los laterales de las fichas coincidan. Se diseñó con el objetivo que sea difícil resolverlo mediante una computadora por fuerza bruta.
URL: <http://www.juegosrox.com/logica/juego-eternity-2.html>
3. **Buscaminas:** El Buscaminas (en inglés: Minesweeper) es un videojuego para un jugador inventado por Robert Donner en 1989. El objetivo del juego es despejar un campo de minas sin detonar ninguna mina.
URL: <http://es.wikipedia.org/wiki/Buscaminas>
4. **GridLock:** Desplazar las piezas en un solo eje para poder sacar el rectángulo azul del tablero.
URL: <http://www.addictinggames.com/puzzle-games/gridlock.jsp>
5. **DeepTrip:** URL: <http://www.minijuegos.com/Deep-Trip/10185>
6. **Mahjong:** El juego Mahjong (también conocido como Taipei) tiene como objetivo eliminar todas las fichas del tablero encontrando los pares iguales. Hacé click sobre cada una de las fichas del par para eliminarlas.
URL: http://www.juegosarea.com/mahjong_solitaire.html
7. **Sokoban:** URL: <http://www.jugarjuegos.com/juegos/java/sokoban3/>
8. **Fill Zone:** Propagá los colores hasta que consigas que todo el tablero este del mismo color. Intentá terminar el nivel antes de que el número de jugadas termine.
URL: http://www.juegosarea.com/fill_zone.html
9. **Edificios:** El tablero representa un espacio, visto desde arriba cubierto de edificios. Los edificios tienen distintas alturas que van del 1 (el más bajo) al 5 (el más alto). Los números que están afuera de la grilla indican cuántos edificios pueden verse al mirar la línea o la columna desde esa dirección. Por ejemplo, en el caso de ejemplo (Figura 1) (hecho con 4 alturas) se ve un 1 en el extremo inferior derecho: éste indica que el primer edificio, de 4 pisos, tapa a todos los demás y es el único que se puede ver. El objetivo del juego es cubrir toda la grilla, cuidando de que en cada línea y en cada columna queden representadas todas las alturas (en un esquema de 5x5 como el del ejercicio -ver Figura 2-, irán del 1 al 5).

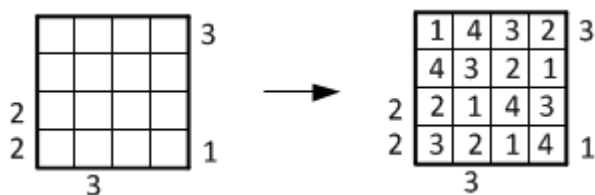


Figura 1: Ejemplo de *Edificios*

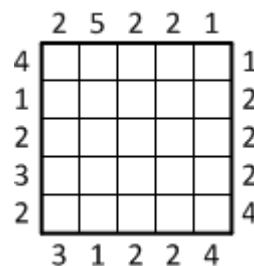


Figura 2: Grilla de 5x5 de *Edificios*