DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

ALGORÍTMICA

ETSIIT

Prácticas

ALGORITMOS DE EXPLORACIÓN EN GRAFOS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL
UNIVERSIDAD DE GRANADA

Objetivos

El objetivo de esta práctica es desarrollar las habilidades para diseñar e implementar algoritmos basados en exploración de grafos. Para ello, se presentará al alumno un conjunto de problemas a solucionar mediante el uso de esta técnica. Cada problema deberá ser resuelto por grupos de alumnos (un problema por grupo).

1. Sudoku

Un Sudoku es un pasatiempo matemático que consiste en rellenar con números del 1 al 9 una tabla de 9x9 elementos, dividida en subcuadrículas de 3x3, con casillas ya rellenadas previamente. Las normas del pasatiempo son:

- No puede haber dos casillas en la misma fila con el mismo número.
- No puede haber dos casillas en la misma columna con el mismo número.
- No puede haber dos casillas, en la misma subcuadrícula de 3x3, con el mismo número.

La siguiente figura muestra un ejemplo de Sudoku sin resolver:

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

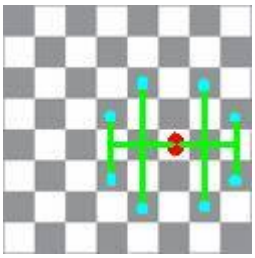
Para solucionar un sudoku, es necesario escribir números en todas las casillas que quedan vacías de modo que cumplan las 3 reglas anteriores. **Las casillas que pueden modificarse son sólo aquellas que al comienzo están vacías**. Por ejemplo, en la columna 3 de la fila 1 podemos poner cualquiera de los números 1, 2, 4. Los números 3 y 5 no pueden escribirse en esta casilla por dos motivos: estar en la misma fila y encontrarse dentro de la misma subregión de 3x3; los números 6 y 9 no se pueden colocar por estar dentro de la misma región de 3x3; el número 7 tampoco es válido porque ya hay una casilla en la misma fila con ese dígito asignado. Por último, el 8

tampoco es válido porque ya existe una casilla en la misma columna que contiene ese dígito.

Se pide diseñar e implementar un algoritmo de exploración en grafos (BackTracking o Branch&Bound) que solucione este problema.

2. El recorrido del caballo

En un tablero de ajedrez, un caballo debe pasar por todas las casillas del tablero sin pisar dos veces la misma casilla y sin hacer ningún movimiento que lo saque del tablero. Sabiendo que un caballo se puede mover en movimientos en L (ver figura), diseñar las componentes de un algoritmo Backtracking o Branch&Bound que resuelva el problema.



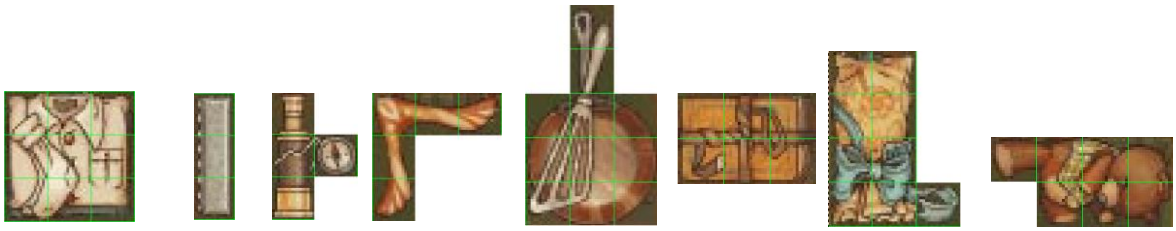
3. El puzzle "La Maleta de Luke" (*El Profesor Layton y la Caja de Pandora, Nintendo DS*)

Fuente de la figura: http://www.guiasnintendo.com

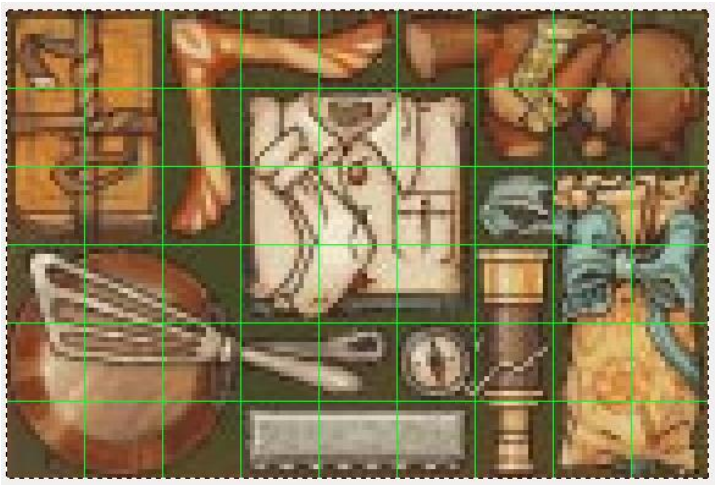
Diversos videojuegos de la categoría puzzle ofrecen rompecabezas que pueden ser estudiados como un problema de exploración en grafos. En particular, el puzzle *"La maleta de Luke"* presentado en *El profesor Layton y la Caja de Pandora* para Nintendo

DS, supone que hay una maleta y diversos objetos que tenemos que introducir en ella sin que se solapen, cada uno de unas dimensiones determinadas.

Estos problemas se pueden discretizar, de modo que en nuestro caso dividiremos la maleta en un rectángulo de 6 casillas de alto por 9 de ancho, y cada objeto también lo discretizaremos con estas mismas dimensiones:



Cada pieza, a la hora de colocarla en una posición de la maleta, puede rotarse 0º, 90º, 180º o 270º para alcanzar la solución. De este modo, el puzzle se resuelve colocando las piezas según la siguiente figura:



Se pide diseñar e implementar un algoritmo de exploración en grafos (BackTracking o Branch&Bound) que solucione este problema con el tamaño de maleta y piezas descritas en este apartado.

4. El problema del mural

Se dispone de una fotografía *F* dividida en *NxM* regiones donde cada región tiene un tamaño de *kxk* píxeles, y un número P (P>= *N*M*) de fotografías p_i más pequeñas $\{p_i, 1<=i<=L\}$ de tamaño *kxk* píxeles. Cada región de *F*, *F(x,y)*, tiene asignado un color entre 0 y 255. A su vez, cada fotografía pequeña p_i también tiene asignado un color entre 0 y 255. El problema consiste en realizar una asignación de cada fotografía p_i a cada región de un mural *F'* de tamaño *NxM*, *F'(x,y),* de modo que la composición del mural $F'(x,y)= p_i$ sea lo más similar posible a la fotografía original *F*; es decir:

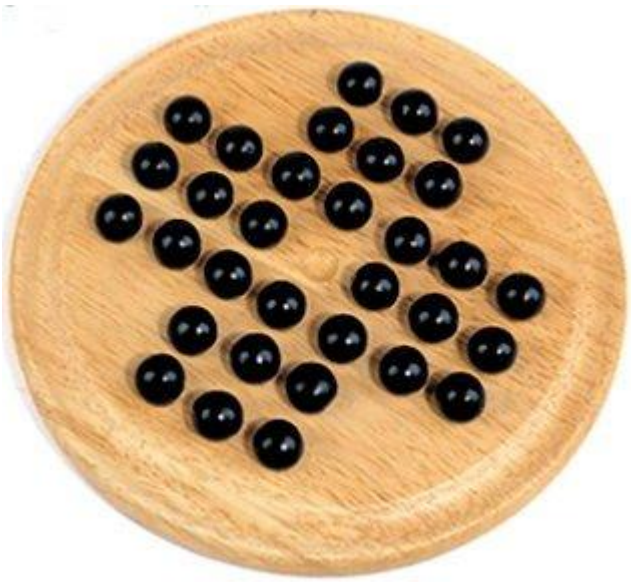
Encontrar una asignación $F'(x,y)= p_i$ tal que se minimice la siguiente función de coste:

$$\sum_{x=1}^{N}\sum_{y=1}^{M}\left|F'(x,y) - F(x,y)\right|$$

Se pide diseñar e implementar un algoritmo BackTracking o Branch&Bound que solucione el problema con el mínimo coste. Como salida, se deberá proporcionar la asignación de la matriz F' a la imagen correspondiente junto con su coste.

5. Senku

En el solitario de mesa llamado Senku (Continental), treinta y dos piezas se colocan en un tablero de treinta y tres casillas tal y como indica la siguiente figura, quedando vacía únicamente la casilla central:



Una pieza sólo puede moverse saltando sobre una de sus vecinas y cayendo en una posición vacía, al igual que en el juego de las damas, aunque aquí no están permitidos los saltos en diagonal (sólo en horizontal y en vertical). La pieza sobre la que se salta se retira del tablero.

El juego finaliza cuando el jugador no puede realizar más movimientos. Dado este caso, si sólo queda una pieza en el tablero, el jugador gana. Si queda más de una pieza, el jugador pierde. Un ejemplo del desarrollo del juego se encuentra en la URL https://www.youtube.com/watch?v=T1AzSqoT8P8 .

Se pide: Desarrollar un algoritmo BackTracking o Branch&Bound que permita resolver el juego de modo que el jugador gane.

6. Enfoque BackTracking/Branch&Bound para resolver los ejercicios

La resolución del ejercicio supone escoger la mejor técnica de exploración para cada problema, seleccionando entre BackTracking y Branch&Bound, **justificando la elección** de la técnica seleccionada frente a la otra técnica no seleccionada. Una vez hecho esto, se deberán diseñar las componentes del algoritmo y adaptar dichas componentes al esqueleto general de la técnica, como parte del diseño y paso previo al desarrollo del código fuente que solucione el problema.

7. Tareas a realizar

Se presentará una memoria de prácticas conteniendo lo siguiente:

1. Se debe incluir un análisis del problema.
2. Escoger una técnica (BackTracking o Branch&Bound) y justificar su elección frente a la otra metodología no seleccionada.
3. Diseño de la solución, describiendo cada una de las componentes de la misma relacionándola con las componentes de un algoritmo BackTracking o Branch&Bound, según la técnica escogida.
4. Esqueleto del algoritmo (pseudocódigo) que soluciona el problema, explicando cómo se ha incorporado cada una de las componentes de diseño de la técnica (BackTracking o Branch&Bound, según se aplique).
5. Explicación del funcionamiento del algoritmo, sobre un caso de ejemplo pequeño propuesto por los estudiantes.

6. Enunciado de un problema o caso real donde se pueda utilizar la técnica seleccionada para solucionarlo (distinto de los comentados en clase).
7. Cálculo del orden de eficiencia teórico del algoritmo.
8. Instrucciones sobre cómo compilar y ejecutar el código de la práctica.

8. Condiciones de entrega

Se deberá entregar, a través de la plataforma online, un fichero ZIP con la siguiente información:

- La solución de los ejercicios planteados en la práctica deberá presentarse en una **memoria de prácticas en formato PDF que contenga, al menos, la respuesta a los apartados requeridos en las tareas a realizar**.
- Se deberá presentar, además, el código fuente de la solución al problema (sólo código fuente, no se aceptarán ejecutables).

- **No se admitirán ficheros .zip que contengan programa ejecutables.**

Adicionalmente:

- En horario de clase, los grupos expondrán su solución ante el resto de la clase. Para ello, se remitirá, mediante la plataforma online docente, un PDF con la presentación, adaptada para su exposición en 5 minutos. Dicha entrega se realizará con posterioridad a la entrega de la memoria, previo aviso por el profesor a través de mensajes por la plataforma docente.

9. Evaluación

Criterios de evaluación:

- La calificación de la práctica será un valor comprendido entre 0 y 10.
- La solución del alumno resuelve correctamente el ejercicio (el algoritmo hace aquello para lo que es diseñado de forma eficiente, limpia y óptima, sin errores, 30% del valor de la pregunta).
- Justificación de la elección de la mejor técnica para resolver el problema (10%).
- Análisis del problema (20%).
- Explicación y diseño de la solución, con ejemplos (40%).