



# Algorítmica

## Grado en Ingeniería Informática 2024

### Guion de prácticas

#### Algoritmos de exploración de grafos

1. Objetivo.....	2
2. Problemas propuestos.....	2
3. Evaluación de la práctica.....	4
4. Entrega y presentación de la práctica.....	4

## Algoritmos de exploración de grafos

### 1. Objetivo

El objetivo de la práctica consiste en que el alumno sea capaz de analizar una serie de problemas y resolverlos mediante las técnicas de diseño de algoritmos basadas en exploración de grafos, como la vuelta atrás (Backtracking) o la ramificación y poda (Branch&Bound)

### 2. Problemas propuestos

Para cada uno de los problemas siguientes, se debe:

- Diseñar e implementar un algoritmo (Backtracking y/o Branch&bound) que resuelva el problema.
- Proponer un ejemplo de uso (instancia del problema) representativo, y explicar cómo funciona el algoritmo sobre este ejemplo, paso a paso.

#### Problema 1

Supongamos que tenemos  $n$  estudiantes en una clase y queremos crear con ellos equipos formados por parejas (podemos suponer que  $n$  es un número par). Se dispone de una matriz  $p$  de tamaño  $n \times n$  en la que  $p(i, j)$  indica el nivel de preferencia que el estudiante  $i$  tiene para trabajar con el estudiante  $j$ . El valor del emparejamiento del estudiante  $i$  con el  $j$  es  $p(i, j) * p(j, i)$ . Se trata de encontrar un emparejamiento para todos los estudiantes de forma que se maximice la suma de los valores de los emparejamientos. Diseñar e implementar un algoritmo de backtracking para resolver el problema.

#### Problema 2

Se va a celebrar una cena de gala a la que asistirán  $n$  invitados. Todos se van a sentar alrededor de una única gran mesa circular, de forma que cada invitado tendrá sentados junto a él a otros dos comensales (uno a su izquierda y otro a su derecha). En función de las características de cada invitado (por ejemplo categoría o puesto, lugar de procedencia,...) existen unas normas de protocolo que indican el nivel de conveniencia de que dos invitados se sienten en lugares contiguos (supondremos que dicho nivel es un número entero entre 0 y 100). El nivel de conveniencia total de una asignación de invitados a su puesto en la mesa es la suma de todos los niveles de conveniencia de cada invitado con cada uno de los dos invitados sentados a su lado. Se desea sentar a los invitados de forma que el nivel de conveniencia global sea lo mayor posible. Diseñar e implementar un algoritmo de backtracking para resolver el problema.

### Problema 3

En el juego (solitario) del senku, también llamado solitario chino, se colocan 32 piezas iguales en un tablero de 33 casillas, tal y como se indica en la siguiente figura (las “x” corresponden a posiciones no válidas):



X	X	O	O	O	X	X
X	X	O	O	O	X	X
O	O	O	O	O	O	O
O	O	O		O	O	O
O	O	O	O	O	O	O
X	X	O	O	O	X	X
X	X	O	O	O	X	X

Solo se permiten movimientos de las piezas en vertical y horizontal. Una pieza solo puede moverse saltando sobre otra y situándose en la siguiente casilla, que debe estar vacía. La pieza sobre la que se salta se retira del tablero. Se consigue terminar con éxito el juego cuando queda una sola pieza en la posición central del tablero (la que estaba inicialmente vacía). Diseñar e implementar un algoritmo de backtracking que encuentre una serie de movimientos para llegar con éxito al final del juego.

En este enlace hay un video de un muchacho ganando el juego:

<https://www.youtube.com/watch?v=1NEnM8FiQmk>

### Problemas 4 y 5

El problema consiste en encontrar la salida de un laberinto. Más concretamente, supondremos que el laberinto se representa mediante una matriz cuadrada bidimensional de tamaño  $n \times n$ . Cada posición almacena un valor *booleano* “true” si la casilla es transitable y “false” si la casilla no es transitable. Los movimientos permitidos son a casillas adyacentes de la misma fila o la misma columna. Podemos suponer que las casillas de entrada y salida del laberinto son  $(0,0)$  y  $(n-1, n-1)$  respectivamente. Por tanto, el problema consiste en, dada una matriz que representa el laberinto, encontrar si existe un camino para ir desde la entrada hasta la salida.

1. Indicar las restricciones (implícitas y explícitas) que nos aseguren un árbol de estados finito para el problema.
2. (**Problema 4**) Diseñar e implementar un algoritmo vuelta atrás para resolver el problema (backtracking).
3. (**Problema 5**) Modificar el algoritmo para que encuentre el camino más corto. En este caso no pararemos cuando encontremos una solución, sino que se seguirá la exploración. No obstante, se puede (y debe) realizar una poda para no explorar soluciones parciales que ya tengan una longitud mayor que la mejor hallada hasta el momento.
4. Ejemplo paso a paso de ambas versiones.

### 3. Evaluación de la práctica

Se deben resolver todos los ejercicios propuestos en el apartado 2 de este guion. Cada ejercicio se valorará sobre 10 de la siguiente forma:

1. (6 puntos) **Diseño** del algoritmo
2. (2 puntos) **Ejemplo** paso a paso de la explicación del funcionamiento del algoritmo para una instancia pequeña propuesta por el grupo de estudiantes.
3. (2 puntos) Correcto funcionamiento de la **implementación**.

La valoración de la práctica se dará como una calificación numérica entre 0 y 10.

### 4. Entrega y presentación de la práctica

Se deberá entregar:

1. Un documento (memoria de prácticas) realizado en equipos, conteniendo la solución a los problemas propuestos, según los apartados descritos en el apartado 2 de este guion.
2. Los ficheros fuente con las diferentes implementaciones, junto con un makefile y un ejemplo de ejecución.

La práctica deberá ser entregada por PRADO, en la fecha y hora límite explicada en clase por el profesor. No se aceptarán, bajo ningún concepto, prácticas entregadas con posterioridad a la fecha límite indicada. La entrega de PRADO permanecerá abierta con, al menos, una semana de antelación antes de la fecha límite, por lo que todo alumno tendrá tiempo suficiente para entregarla.

El profesor, en clase de prácticas, realizará controles de las prácticas a discreción, que consistirán en presentaciones de los estudiantes de cada equipo (powerpoint) y/o entrevistas individuales con el fin de asegurar de que los estudiantes alcanzan las competencias deseadas. Estas entrevistas y/o presentaciones se realizarán en las sesiones de evaluación de prácticas, previamente anunciadas en clase por el profesor.

La **no asistencia** a una sesión de evaluación de prácticas por un estudiante supondrá la **calificación de 0 (no presentado) a la práctica que deba presentar, independientemente de la calificación obtenida en la memoria de prácticas**.

**IMPORTANTE:** Antes de las sesiones de evaluación, cada estudiante deberá prepararse para:

- Realizar una presentación powerpoint de la práctica, si el profesor lo exige.
- Conocer a fondo todos los algoritmos resueltos, así como los pasos para diseñar e implementar los algoritmos.
- Conocer las respuestas a las preguntas requeridas en el apartado 3 para cada problema.