Trabajo práctico 4 - Backtracking

Programación 3 - TUDAI

Implemente algoritmos por la técnica Backtracking para los siguientes problemas.

Ejercicio 1.
Se tiene un conjunto de salas comunicadas entre sí a través de puertas que se abren solamente en un sentido. Una de las salas se denomina entrada y la otra salida. Construir un algoritmo que permita ir desde la entrada a la salida atravesando la máxima cantidad de salas. Idea: podría representar el problema mediante un grafo dirigido, donde cada nodo es una habitación, y cada puerta es un arco dirigido hacia otra habitación.

Ejercicio 2.
Dado un laberinto consistente en una matriz cuadrada que tiene en cada posición un valor natural y cuatro valores booleanos, indicando estos últimos si desde esa casilla se puede ir al norte, este, sur y oeste, encontrar un camino de longitud mínima entre dos casillas dadas, siendo la longitud de un camino la suma de los valores naturales de las casillas por las que pasa. Idea: podría representarse el laberinto como una matriz, de objetos, donde cada objeto tiene el valor natural, y cuatro booleanos, uno para cada dirección a la que se permite ir desde allí.

Ejercicio 3.
Suma de subconjuntos. Dados n números positivos distintos, se desea encontrar todas las combinaciones de esos números tal que la suma sea igual a M.

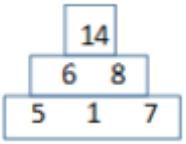
Ejercicio 4.
Partición de conjunto. Dado un conjunto de n enteros se desea encontrar, si existe, una partición en dos subconjuntos disjuntos, tal que la suma de sus elementos sea la misma.

Ejercicio 5.
Asignación de tareas a procesadores. Se tienen m procesadores idénticos y n tareas con un tiempo de ejecución dado. Se requiere encontrar una asignación de tareas a procesadores de manera de minimizar el tiempo de ejecución del total de tareas.

Ejercicio 6.
Caballo de Atila. Por donde pisa el caballo de Atila jamás vuelve a crecer el pasto. El caballo fue directamente hacia el jardín de n x n casillas. Empezó su paseo por una casilla cualquiera y volvió a ella, es decir hizo un recorrido cerrado. No visitó dos veces una misma casilla, se movió de una casilla a otra vecina en forma horizontal o vertical, pero nunca en diagonal. Por donde pisó el caballo, el pasto jamás volvió a crecer. Luego de terminado el recorrido en algunas casillas todavía había pasto (señal de que en ellas no había estado el caballo). Escriba un algoritmo que deduzca el recorrido completo que hizo el caballo.

Ejercicio 7.
Tablero mágico. Dado un tablero de tamaño n x n, construir un algoritmo que ubique (si es posible) n*n números naturales diferentes, entre 1 y un cierto k (con k>n*n), de manera tal que la suma de las columnas y de las filas sea igual a S.

Ejercicio 8.
Colocar un entero positivo (menor que un cierto valor entero k dado) en cada casilla de una pirámide de base B (valor entero dado) de modo que cada número sea igual a la suma de las casillas sobre las que está apoyado. Los números de todas las casillas deben ser diferentes.



Ejercicio 9.
Dado un tablero de 4 x 4, en cuyas casillas se encuentran desordenados los números enteros del 1 al 15 y una casilla desocupada en una posición inicial dada, determinar una secuencia de pasos tal intercambiando números contiguos (en horizontal y en vertical) con la casilla desocupada, los números en el tablero queden ordenados (como muestra la figura) y la casilla desocupada quede en la posición 4,4.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Ejercicio 10
Utilizando la técnica Backtraking, escriba un algoritmo que dado un conjunto de números enteros, devuelva (si existen) todos los subconjuntos de tamaño N (dado como parámetro), cuyas sumas sean exactamente cero.
Por ejemplo dado el conjunto {-7, -3, -2, -1, 5, 8 } y N = 3, los subconjuntos que suman cero son: {-7, -1, 8} y {-3, -2, 5}.

Ejercicio 11
El robot de limpieza necesita volver desde su posición actual hasta su base de carga. Dado que al robot le queda poca batería, desea encontrar el camino más corto. El robot dispone de un mapa de la casa representado como una matriz, donde cada celda es una posición de la casa. La matriz posee un 0 si la celda está vacía, o un 1 si la celda presenta algún obstáculo (por ejemplo, un mueble). Se desea encontrar entonces el camino más corto considerando que:

- Desde una celda solo te puedes mover a las celdas contiguas (izquierda, derecha, arriba y abajo)
- El robot sólo puede caminar por celdas libres (no por celdas con obstáculos)

¿Hay alguna poda que se pueda aplicar al algoritmo?