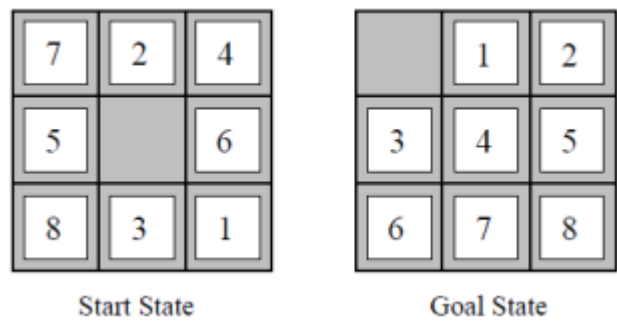


# SI, TGR 1

## Ejercicio 1, Puzzle 3x3

El 8-puzzle consiste en un tablero de 3x3 con ocho fichas numeradas y un espacio en blanco. Una ficha adyacente al espacio en blanco puede deslizarse a éste. La meta es alcanzar el estado objetivo especificado tal como se muestra en la figura de la derecha en el menor número de movimientos posible.



- **Estado inicial:** Cualquiera con una matriz con valores del 0 al 8 sin repeticiones, donde el 0 es el hueco
- **Conjunto de acciones**

Acción	Precondiciones
Mover arriba	Debe estar el hueco arriba
Mover abajo	Debe estar el hueco abajo
Mover izquierda	Debe estar el hueco a la izquierda
Mover derecha	Debe estar el hueco a la derecha

- **Modelo de transiciones**

Acción	Precondición	Resultado
Mover arriba	Debe estar el hueco arriba	Intercambiar casilla con el hueco
Mover abajo	Debe estar el hueco abajo	Intercambiar casilla con el hueco
Mover izquierda	Debe estar el hueco a la izquierda	Intercambiar casilla con el hueco
Mover derecha	Debe estar el hueco a la derecha	Intercambiar casilla con el hueco

- **Prueba de meta:** El array está ordenado de izquierda a derecha y de arriba a abajo.
- **Función coste de camino:** Sumar número de movimientos desde el estado inicial hasta llegar al final.

## TGR 2, Formas de búsqueda

Estrategia básica 2

```

1  1. S_actual = S_inicial
2    E = ∅
3  2. Prueba de meta
4    Si S_actual es meta → Fin
5    Si no → Seguir
6  3. a = siguiente acción aplicable a S_actual
7  4. Contains(E, a)
8    Si -> volver a 3
9    No -> S_actual = a;
10   Insert(a, E)
11  5. Volver a 2

```

**a. Problema: 8-puzzle.**

**Algoritmo:** Estrategia de búsqueda básica 2 (modificada para registrar acciones)

**Estado inicial / meta:**

1	2	
3	4	5
6	7	8

**Start State**

	1	2
3	4	5
6	7	8

**Goal State**

El algoritmo no da la solución porque queda en el mismo estado, muere de inanición al solo poder hacer 4 movimientos.

Paso	S <sub>actual</sub>	Siguiente acción
1	[1 2 _] [3 4 5] [6 7 8]	Arriba
2	[1 2 5] [3 4 _] [6 7 8]	Abajo
3	[1 2 _] [3 4 5] [6 7 8]	Derecha
4	[1 _ 2] [3 4 5] [6 7 8]	Izquierda
5	[1 2 _] [3 4 5] [6 7 8]	

b. Problema: 8-puzzle

Algoritmo: Estrategia de búsqueda básica 4 (modificada para registrar estados explorados)

Estado inicial / meta:

1	2	
3	4	5
6	7	8

Start State

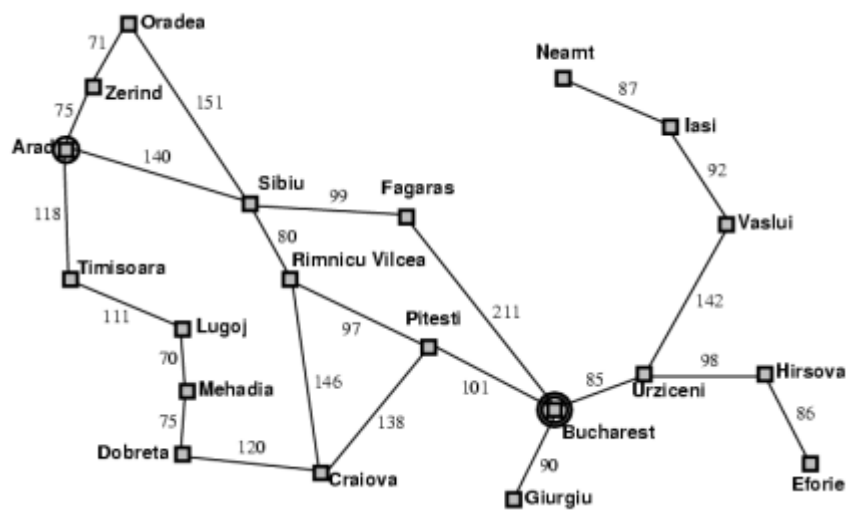
	1	2
3	4	5
6	7	8

Goal State

Paso	S <sub>actual</sub>	Acciones -> Estados
1	[1 2 _] [3 4 5] [6 7 8]	Arriba -> 1 2 Derecha -> 1 3
2	[1 2 5] [3 4 _] [6 7 8]	Arriba -> 1 2 4 Abajo -> 1 2 1, No se puede hacer porque repetimos estado Derecha -> 1 2 5
3	[1 _ 2] [3 4 5] [6 7 8]	Arriba -> 1 3 6 Izquierda -> 1 3 1, No se puede realizar porque repetimos estado Derecha -> 1 3 7
4	[1 2 5] [3 4 8] [6 7 _]	Abajo -> 1 2 4 2, No se puede realizar porque repetimos estado Derecha -> 1 2 4 ...
5	[1 2 5] [3 _ 4] [6 7 8]	Arriba -> 1 2 5 ... Abajo -> 1 2 5 ... Izquierda -> 1 2 5 ... Derecha -> 1 2 5 2, No se puede realizar porque repetimos estado
6	[1 4 2] [3 _ 5] [6 7 8]	Arriba -> 1 3 6 ... Abajo -> 1 3 6 3, No se puede realizar porque repetimos estado Izquierda -> 1 3 6 ... Derecha -> 1 3 6 ...
7	[_ 1 2] [3 4 5] [6 7 8]	Solución: 1 3 7

## Ejercicio 2, Viaje de Arad a Bucarest

Mapa de Rumanía simplificado, que incluye los costes de los tramos individuales. El objetivo es ir desde Arad a Bucarest. Podemos suponer que el mapa viene representado como una serie de tuplas (ciudad origen, ciudad destino, km) que representan las carreteras (ejemplo: (Arad, Sibiu, 140)).



- **Estado inicial:** Un nodo de origen, uno de destino y un grafo con peso conexo al que pertenecen.
- **Conjunto de acciones:**

Acción	Precondición
Moverse a un nodo	Nodo no visitado

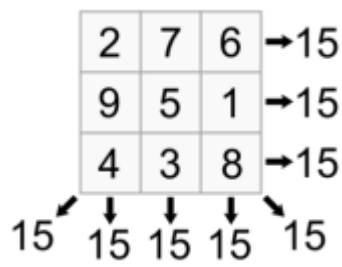
- **Modelo de transiciones**

Acción	Precondición	Resultado
Moverse a un nodo	Nodo no visitado	(nodoOrigen, nodoDestino, peso + pesoAcumulado) nodoOrigen = nodoDestino nodoDestino = nuevoNodoDestino

- **Prueba de meta:** Comprobar que nodoDestino sea el nodo deseado.
- **Función coste de camino:** Retornar pesoAcumulado

### Ejercicio 3, Cuadrado mágico $N \times N$

Un cuadrado mágico de  $N \times N$  es una matriz que contiene los números entre 1 y  $N^2$  dispuestos de tal manera que la suma de los elementos de cada una de sus filas (o de sus columnas o de sus diagonales principales) es siempre la misma:  $\frac{N(N^2+1)}{2}$



- **Estado inicial:** Matriz NxN vacía
- **Conjunto de acciones:**

Acción	Precondición
Insertar número	Número no repetido Casilla vacía Suma de fila, columna $\leq \frac{N(N^2+1)}{2}$ Si está en diagonal, la suma $\leq \frac{N(N^2+1)}{2}$
Borrar número	Halla casillas vacías y no se puedan insertar más números

- **Modelo de transiciones:**

Acción	Precondición	Resultado
Insertar número	Número no repetido Casilla vacía Suma de fila y columna $\leq \frac{N(N^2+1)}{2}$ Si está en diagonal, la suma $\leq \frac{N(N^2+1)}{2}$	$\text{cuadrado}[x][y] = \text{nuevoNumero}$
Borrar número	Halla casillas vacías y no se puedan insertar más números	$\text{cuadrado}[x][y] = 0$ Descartar número para esa casilla

- **Prueba de meta:** Suma de todas las filas, columnas y diagonales debe ser igual a  $\frac{N(N^2+1)}{2}$
- **Función coste de camino:** Número de casillas insertadas o eliminadas en total.

# Ejercicio 4, Sudoku

El sudoku es un pasatiempo cuyo objetivo es rellenar una cuadrícula de 9x9 celdas dividida en subcuadrículas de 3x3 con las cifras del 1 al 9 partiendo de algunos números ya dispuestos en algunas de las celdas. No se debe repetir ningún número en una misma fila, columna o subcuadrícula.

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

- **Estado inicial:** Matriz 9x9 con algunas casillas ya rellenas
- **Conjunto de movimientos**

Acción	Precondición
Insertar número	Número no repetido en fila, columna o subcuadrante Casilla vacía
Borrar número	Halla casillas y no se puedan insertar más números

- **Modelo de transiciones**

Acción	Precondición	Resultado
Insertar número	Número no repetido en fila, columna o subcuadrante Casilla vacía	cuadrado[x][y] = nuevoNumero
Borrar número	Halla casillas vacías y no se puedan insertar más números	cuadrado[x][y] = 0 Descartar número para esa casilla

- **Prueba de meta:** Matriz con filas, columnas y subcuadrantes con números del 1 al 9
- **Función coste de camino:** Número de casillas insertadas y borradas en total