

BACKTRACKING: SUDOKU

Juan Antonio Betancourt

¿Que es un sudoku?

Es un juego, el objetivo es llenar todas las casillas de numeros.

Se desarrolla en una matriz de 9x9, con sub-bloques de 3x3, (3 bloques de 3x3)

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		

Reglas para ganar

- • • • •
- • • • •
- • • • •

1. No puede haber un numero repetido en las filas u columnas
2. Cada sub-bloque de 3x3 no puede tener un numero repetido

03

Problema

Problema 3

Diseño de su solución

Problema 2

Constatar que la
solución numérica de
una casilla cumple las
condiciones

Problema 1

Codificar su solución
haciendo uso de
backtracking y
recursividad

Soluciones

Solution 1

Hacer uso de ciclos for para simplificar el código

Solution 2

crear una función auxiliar para constatar en cada llamada que se cumplen las condiciones

Solution 3

hacer uso del lenguaje de java y su api para codificar la solución

Problema 3: Diseño de solución

Lo primero es representar un sudoku en una matriz 9x9 en java.

Los ceros representan el estado de vacío, así se sabe que es un punto que se puede llenar

{5, 3, 0, 0, 7, 0, 0, 0, 0}
{6, 0, 0, 1, 9, 5, 0, 0, 0}
{0, 9, 8, 0, 0, 0, 0, 6, 0}
{8, 0, 0, 0, 6, 0, 0, 0, 3}
{4, 0, 0, 8, 0, 3, 0, 0, 1}
{7, 0, 0, 0, 2, 0, 0, 0, 6}
{0, 6, 0, 0, 0, 0, 2, 8, 0}
{0, 0, 0, 4, 1, 9, 0, 0, 5}
{0, 0, 0, 0, 8, 0, 0, 7, 9}



Pasos lógicos

1. Recorrer horizontalmente hasta hallar un cero o agotar los valores
 - 1.1 al encontrar un 0, se probaran valores del 0 al 9
2. Verificar que el valor agregado cumpla las condiciones.
 - 2.1 Que no este en la fila ni columna.
 - 2.2 Que en su subgrupo no este repetido.

Pasos lógicos

```
{5, 3, 0, 0, 7, 0, 0, 0, 0}  
{6, 0, 0, 1, 9, 5, 0, 0, 0}  
{0, 9, 8, 0, 0, 0, 0, 6, 0}  
{8, 0, 0, 0, 6, 0, 0, 0, 3}  
{4, 0, 0, 8, 0, 3, 0, 0, 1}  
{7, 0, 0, 0, 2, 0, 0, 0, 6}  
{0, 6, 0, 0, 0, 0, 2, 8, 0}  
{0, 0, 0, 4, 1, 9, 0, 0, 5}  
{0, 0, 0, 0, 8, 0, 0, 7, 9}
```

0=1

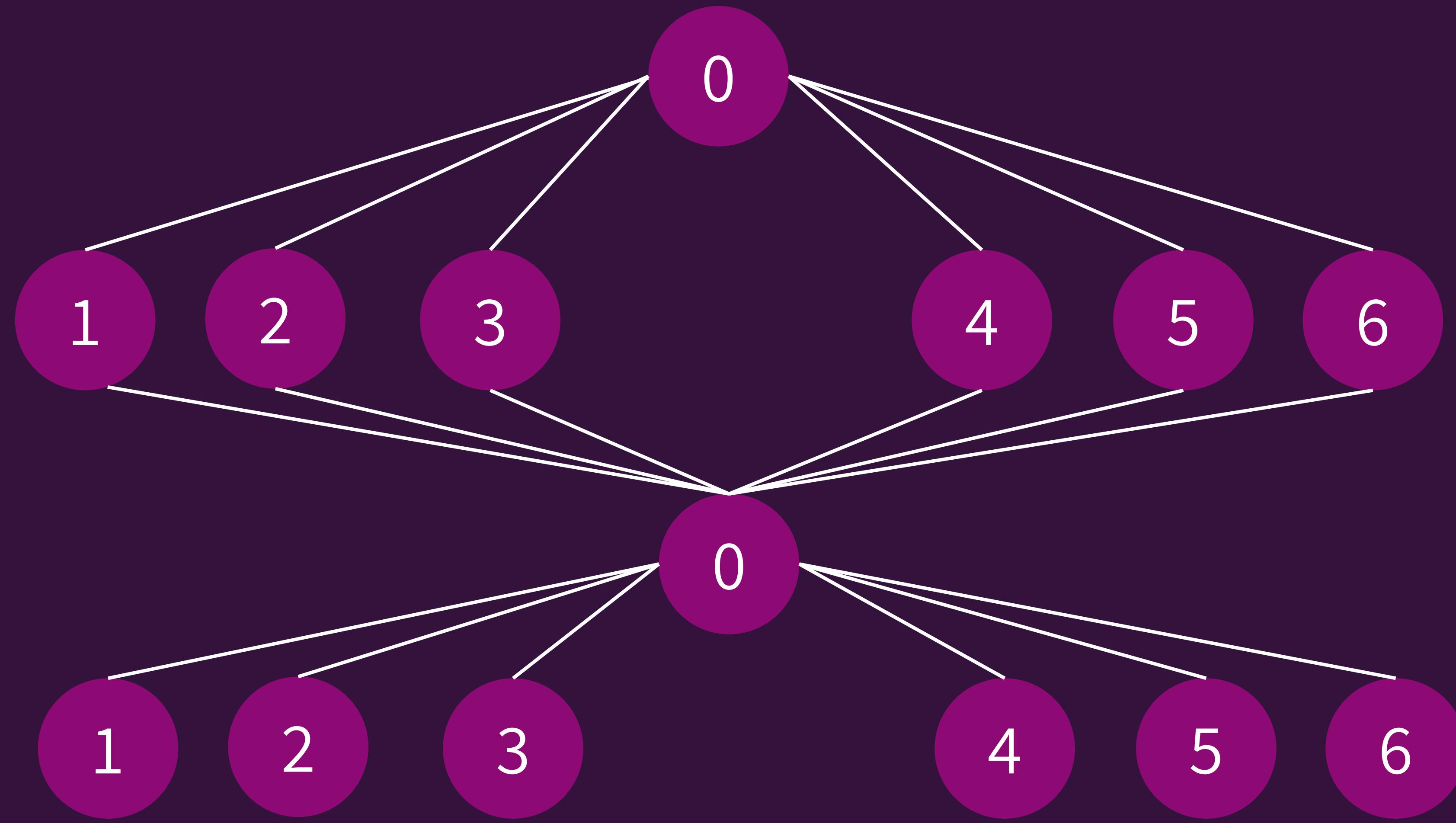
Pasos lógicos

{5, 3, 0, 0, 7, 0, 0, 0, 0}
{6, 0, 0, 1, 9, 5, 0, 0, 0}
{0, 9, 8, 0, 0, 0, 0, 6, 0}
{8, 0, 0, 0, 6, 0, 0, 0, 3}
{4, 0, 0, 8, 0, 3, 0, 0, 1}
{7, 0, 0, 0, 2, 0, 0, 0, 6}
{0, 6, 0, 0, 0, 0, 2, 8, 0}
{0, 0, 0, 4, 1, 9, 0, 0, 5}
{0, 0, 0, 0, 8, 0, 0, 7, 9}

Pasos lógicos

{5, 3, 1, 0, 7, 0, 0, 0, 0},
{6, 0, 0, 1, 9, 5, 0, 0, 0},
{0, 9, 8, 0, 0, 0, 0, 6, 0},
{8, 0, 0, 0, 6, 0, 0, 0, 3},
{4, 0, 0, 8, 0, 3, 0, 0, 1},
{7, 0, 0, 0, 2, 0, 0, 0, 6},
{0, 6, 0, 0, 0, 0, 2, 8, 0},
{0, 0, 0, 4, 1, 9, 0, 0, 5},
{0, 0, 0, 0, 8, 0, 0, 7, 9}

ARBOL DE SOLUCIONES



CODIFICACION