

## Algoritmos y Estructura de Datos I

Primer cuatrimestre de 2017

1 de Abril de 2017

## TPE - Sudoku

El Sudoku es un juego matemático que se publicó por primera vez a finales de la década de 1970 y se popularizó en Japón en 1986, dándose a conocer en el ámbito internacional en 2005 cuando numerosos periódicos empezaron a publicarlo en su sección de pasatiempos. 1 El objetivo del sudoku es rellenar una cuadrícula de  $9 \times 9$  celdas (81 casillas) dividida en subcuadrículas de  $3 \times 3$  (también llamadas “cajas” o “regiones”) con las cifras del 1 al 9 partiendo de algunos números ya dispuestos en algunas de las celdas.

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

Para poder considerar un tablero de Sudoku como **válido** debe cumplir las siguientes condiciones:

- Ser una matriz de  $9 \times 9$
- Todos los elementos de la matriz son los dígitos entre 1 y 9, más el valor 0 que se interpreta como celda vacía.

Decimos que un tablero de Sudoku está **parcialmente resuelto** si cumple las siguientes condiciones:

- Es un tablero válido
- Ninguna fila debe tener un dígito repetido entre 1 y 9 (el 0 se ignora)
- Ninguna columna debe tener un dígito repetido entre 1 y 9 (el 0 se ignora)
- Ninguna región debe tener un dígito repetido entre 1 y 9 (el 0 se ignora)

Decimos que un tablero de Sudoku está **totalmente resuelto** si cumple las siguientes condiciones

- Es un tablero parcialmente resuelto
- No hay ninguna celda vacía (ninguna celda contiene el 0)

Dado que el tablero de Sudoku se representa como una matriz cuadrada de  $9 \times 9$  usando el tipo  $seq\langle seq\langle \mathbb{Z} \rangle \rangle$ , donde:

- Cada posición de la secuencia de secuencias representa una fila de la matriz.
- Ejemplo:  $m[i][j]$  es el elemento en la  $i$ -ésima fila y la  $j$ -ésima columna de la matriz.

Se desean especificar formalmente (i.e. indicar precondition y postcondition) de las siguientes operaciones:

1. **proc sudoku\_esTableroValido**(in t:  $seq\langle seq\langle \mathbb{Z} \rangle \rangle$ , out result: Bool):

- Retorna **True** sii el tablero es un tablero válido.

2. **proc sudoku\_esCeldaVacía**(in t:  $seq\langle seq\langle \mathbb{Z} \rangle \rangle$ , in f:  $\mathbb{Z}$ , in c:  $\mathbb{Z}$ , out result: Bool):

- Dado un tablero  $t$  válido,  $f$  es una fila válida (entre 0 y 8) y  $c$  es una columna válida (entre 0 y 8)
  - Retorna *true* sii la celda en la fila  $f$  y la columna  $c$  está vacía (contiene un 0).
3. **proc sudoku\_nroDeCeldasVacias(in t: seq<seq<Z>>, out result: Z):**
- Dado un tablero  $t$  válido,
  - retorna la cantidad de celdas vacías (o el valor 0 si el tablero está totalmente resuelto).
4. **proc sudoku\_primeraCeldaVacíaFila(in t: seq<seq<Z>>, out result: Z):**
- Dado un tablero  $t$  válido, retorna la fila de la primera celda vacía (o el valor  $-1$  si no hay celdas vacías).
  - La *primera* celda vacía es aquella que cumple las siguientes condiciones:
    - Su número de fila es menor o igual a todos los números de filas de todas las celdas vacías.
    - De todas las celdas con menor número de fila, su número de columna es menor o igual a todas las celdas libres en esa fila.
5. **proc sudoku\_primeraCeldaVacíaColumna(in t: seq<seq<Z>>, out result: Z):**
- Dado un tablero  $t$  válido, retorna la columna de la primera celda vacía (o el valor  $-1$  si no hay celdas vacías).
  - La *primera* celda vacía es aquella que cumple las siguientes condiciones:
    - Su número de fila es menor o igual a todos los números de filas de todas las celdas vacías.
    - De todas las celdas con menor número de fila, su número de columna es menor o igual a todas las celdas libres en esa fila.
6. **proc sudoku\_valorEnCelda(in t: seq<seq<Z>>, in f: Z, in c: Z, out result: Z):**
- $t$  es un tablero válido,  $f$  es una fila válida (entre 0 y 8) y  $c$  es una columna válida (entre 0 y 8)
  - La celda en la fila  $f$  y la columna  $c$  no está vacía.
  - Retorna el valor contenido en la fila  $f$  y la columna  $c$ .
7. **proc sudoku\_llenarCelda(inout t: seq<seq<Z>>, in f: Z, in c: Z, in value: Z):**
- $t$  es un tablero válido,  $f$  es una fila válida (entre 0 y 8) y  $c$  es una columna válida (entre 0 y 8)
  - La celda en la fila  $f$  y la columna  $c$  está vacía.
  - *value* es un valor entre 1 y 9
  - Modifica el valor de la fila  $f$  y la columna  $c$  con el valor *value*.
8. **proc sudoku\_vaciarCelda(inout t: seq<seq<Z>>, in f: Z, in c: Z):**
- $t$  es un tablero válido,  $f$  es una fila válida (entre 0 y 8) y  $c$  es una columna válida (entre 0 y 8)
  - La celda en la fila  $f$  y la columna  $c$  **no** está vacía.
  - Elimina el valor que estaba contenido en la celda en la fila  $f$  y la columna  $c$ .
9. **proc sudoku\_esTableroParcialmenteResuelto(in t: seq<seq<Z>>, out result: Bool):**
- Retorna **True** sii el tablero es un tablero parcialmente resuelto.
10. **proc sudoku\_esTableroTotalmenteResuelto(in t: seq<seq<Z>>, out result: Bool):**
- Dado un tablero válido retorna **True** sii el tablero está totalmente resuelto
11. **proc sudoku\_esSubTablero(in t0,t1: seq<seq<Z>>, out result: Bool):**
- Dados  $t_0$  y  $t_1$  dos tableros válidos.
  - Retorna **True** sii todas las celdas no vacías del tablero  $t_0$  tienen el mismo valor en el tablero  $t_1$ .
12. **proc sudoku\_tieneSolucion(in t: seq<seq<Z>>, out tieneSolucion: Bool):**
- Dado un tablero  $t$  válido.
  - Retorna *true* sii las celdas vacías del tablero actual pueden llenarse y resulta en un tablero totalmente resuelto.
13. **proc sudoku\_resolver(inout t: seq<seq<Z>>, out tieneSolucion: Bool):**
- Si el tablero  $t$  tiene solución, completa el tablero con alguna solución posible y retorna *true*.
  - Caso contrario, no modifica el tablero y retorna *false*
14. **proc sudoku\_copiarTablero(in src: seq<seq<Z>>, out target: seq<seq<Z>>):**
- Copia el contenido del tablero *src* en el tablero *target*.