

CS2023 - Aula de Ejercicios N° 4
Brenner H. Ojeda Rios
Semestre 2024-0

Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo.

Ejercicios

1. (5 pts) Dada una arreglo de números enteros *nums* y un número entero *k*, devuelve los *k* elementos más frecuentes. Puede devolver la respuesta en cualquier orden. La complejidad temporal de su algoritmo debe ser mejor que $O(n \log n)$, donde *n* es el tamaño de el arreglo.

■ **Ejemplo 1:**

Input:

```
nums = [1,1,1,2,2,3], k = 2
```

Output:

```
[1,2]
```

■ **Ejemplo 2:**

Input: nums = [1], k = 1

Output: [1]

Restricciones:

- $1 \leq \text{nums.length} \leq 10^5$
- $-10^4 \leq \text{nums}[i] \leq 10^4$
- *k* está en el rango [1, el número de elementos únicos en el arreglo].
- Se garantiza que la respuesta es única.

2. (6 pts) Dada una serie de cadenas, agrupe los anagramas. Puede devolver la respuesta en cualquier orden. Un anagrama es una palabra o frase formada reorganizando las letras de una palabra o frase diferente, normalmente usando todas las letras originales exactamente una vez.

■ **Ejemplo 1:**

Input:

```
strs = ["eat","tea","tan","ate","nat","bat"]
```

Output:

```
[["bat"],["nat","tan"],["ate","eat","tea"]]
```

■ **Ejemplo 2:**

Input:

```
strs = []
```

Output:

```
[[""]]
```

Restricciones:

- $1 \leq \text{strs.length} \leq 10^4$
- $0 \leq \text{strs}[i].length \leq 100$
- *strs[i]* consta de letras minúsculas en inglés.

3. (9 pts) Determina si un tablero de Sudoku de 9 x 9 es válido. Sólo es necesario validar las celdas llenas de acuerdo con las siguientes reglas:

- Cada fila debe contener los dígitos del 1 al 9 sin repetición.
- Cada columna debe contener los dígitos del 1 al 9 sin repetición.
- Cada uno de los nueve subcuadros de 3 x 3 de la cuadrícula debe contener los dígitos del 1 al 9 sin repetición.

Nota:

- Un tablero de Sudoku (parcialmente lleno) podría ser válido pero no necesariamente tiene solución.
- Sólo es necesario validar las celdas completadas de acuerdo con las reglas mencionadas.

Ejemplo:

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

Input: board =

```
[["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"]]
```

Output: true

Restricciones:

- board.length = 9
- board[i].length = 9
- board[i][j] es un dígito 1-9 or '.'.