

CS2023 - Aula de Ejercicios N° 8
Brenner Ojeda
Semestre 2024-1

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

IMPORTANTE:

- Enviar en canvas un **único archivo .cpp** (colocar las 3 soluciones en el mismo archivo .cpp).
- **No se revisaran archivos .zip** subidos a canvas.
- **Colocar nombres y apellidos** de los integrantes **al inicio** del archivo .cpp.
- La evaluación continua es grupal (mínimo 2 alumnos y máximo 3 alumnos).

Ejercicios

1. (6 pts) Se te da una matriz binaria $m \times n$ llamada **mat** compuesta de 1's (que representan soldados) y 0's (que representan civiles). Los soldados están posicionados delante de los civiles. Es decir, todos los 1's aparecerán a la izquierda de todos los 0's en cada fila.

Una fila i es más débil que una fila j si se cumple una de las siguientes condiciones:

- El número de soldados en la fila i es menor que el número de soldados en la fila j .
- Ambas filas tienen el mismo número de soldados y $i < j$.

Devuelve los índices de las k filas más débiles en la matriz ordenadas de la más débil a la más fuerte.

■ **Ejemplo 1:**

Input:

```
mat = [[1,1,0,0,0],  
       [1,1,1,1,0],  
       [1,0,0,0,0],  
       [1,1,0,0,0],  
       [1,1,1,1,1]]  
      ,k = 3
```

Output: [2,0,3]

Explicación: El número de soldados en cada fila es: Fila 0: 2, Fila 1: 4, Fila 2: 1, Fila 3: 2, Fila 4: 5. Las filas ordenadas de la más débil a la más fuerte son [2, 0, 3, 1, 4].

■ **Ejemplo 2:**

```
Input: mat = [[1,0,0,0],  
              [1,1,1,1],  
              [1,0,0,0],  
              [1,0,0,0]]  
            ,k = 2
```

Output: [0,2]

Restricciones:

- $m = \text{mat.length}$ y $n = \text{mat}[i].\text{length}$
- $1 \leq k \leq m$
- $2 \leq n, m \leq 100$
- $\text{mat}[i][j] \in \{0, 1\}$

2. (7 pts) Diseñe una clase para encontrar el k-esimo elemento más grande en una secuencia.

Clase de implementación KthLargest:

- KthLargest(int k, int[] nums) Inicializa el objeto con el número entero k y la secuencia de números enteros nums.
- int add(int val) Agrega el número entero val a la secuencia y devuelve el elemento que representa el k-esimo elemento más grande de la secuencia.

Nota: Su implementación debe ser lo más óptima posible, no se aceptarán soluciones de fuerza bruta. **Pista:** use heap

■ **Ejemplo 1:**

Input:

```
["KthLargest", "add", "add", "add", "add", "add"]  
[[3, [4, 5, 8, 2]], [3], [5], [10], [9], [4]]
```

Output:

```
[null, 4, 5, 5, 8, 8]
```

Explicación:

```
KthLargest kthLargest = new KthLargest(3, [4, 5, 8, 2]);  
kthLargest.add(3);    // return 4  
kthLargest.add(5);    // return 5  
kthLargest.add(10);   // return 5  
kthLargest.add(9);    // return 8  
kthLargest.add(4);    // return 8
```

3. (7 pts) Dada una matriz $n \times n$ donde cada una de las filas y columnas está ordenada en orden ascendente, devuelve el k-ésimo elemento más pequeño de la matriz.

Nota que es el k-ésimo elemento más pequeño en el *orden ordenado*, no el k-ésimo elemento distinto.

Debes encontrar una solución con una complejidad de memoria mejor que $O(n^2)$.

■ **Ejemplo 1:**

Input: matrix = [[1,5,9],[10,11,13],[12,13,15]], k = 8

Output: 13

Explicación: Los elementos en la matriz son [1, 5, 9, 10, 11, 12, 13, 13, 15], y el octavo número más pequeño es 13.

■ **Ejemplo 2:**

Input: matrix = [[-5]], k = 1

Output: -5

Restricciones:

- $n = \text{matrix.length} = \text{matrix}[i].\text{length}$
- $1 \leq n \leq 300$ y $1 \leq k \leq n^2$
- $-10^9 \leq \text{matrix}[i][j] \leq 10^9$
- Todas las filas y columnas de la matriz están garantizadas de estar ordenadas en orden no decreciente.