

# UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

Curso: Algoritmos y Estructuras de Datos-CC182

Evaluación: Práctica Calificada 1

Docente: Walter Cueva Chávez

Sección: SS31

Duración: 110 minutos

Inicio: 1:00 P.M.

## Consideraciones

- Sus respuestas de los ejercicios deben desarrollarlo en Google Colab.
- El envío tiene que ser enviado como archivo con extensión “ipynb” como archivo adjunto en el enlace del aula virtual.
- Todo intento de plagio parcial o total es penalizado con una calificación de cero (0).
- Cada ejercicio debe tener un caso de prueba.
- Solamente implemente los métodos solicitados.






## Desarrollo

### 1. Análisis de Complejidad y Templates

Una matriz bidimensional de enteros(short) como éste:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Pero en memoria se almacena como un puntero a un arreglo de punteros de arreglos unidimensionales.

OABC					
					
A000		1	2	3	4
A008		5	6	7	8
A010		9	10	11	12
A018		13	14	15	16

Por tanto, es posible registrarlo en un solo arreglo unidimensional como

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

De esto concluimos que:

- `matriz[0][0]` es igual al `arreglo[0]`
- ...
- `matriz[1][1]` es igual al `arreglo[5]`

Se solicita:

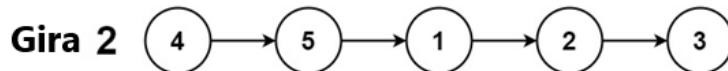
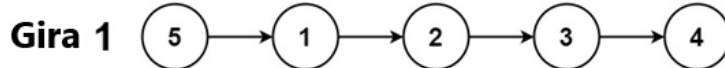
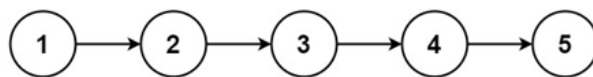
1. (2 puntos) Implementar una estructura de datos llamado "Matrix2d", con uso de templates que implemente un arreglo unidimensional, las dimensiones(m y n) de la matriz son atributos de la clase.
2. (3 puntos) Implementar un método que simule el funcionamiento de una eliminación en una matriz bidimensional, pero utilizando un arreglo unidimensional.
3. (2 puntos) Implementar un método que muestre todos los elementos dado como parámetro el id de la columna determinada.
4. (3 puntos) Realizar su análisis de complejidad detallado y Big O de su método implementado.

## 2. Listas Simple, Archivos, Recursividad

Dado un archivo "input.txt" que tiene una fila de números en el rango [0, 100].

Dada una lista enlazada, girar la lista hacia la derecha en k posiciones.

**Ejemplo 1:**

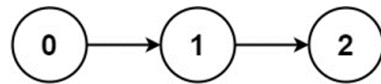


Entrada Archivo: [1,2,3,4,5]

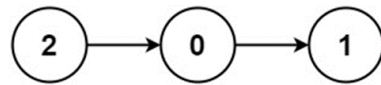
Entrada Teclado: k = 2

Salida: [4,5,1,2,3]

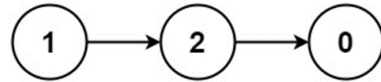
**Ejemplo 2:**



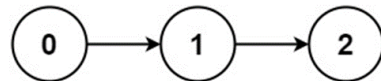
**Gira 1**



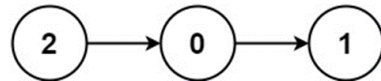
**Gira 2**



**Gira 3**



**Gira 4**



```
Entrada archivo:[0,1,2]  
Entrada teclado: k = 4  
Salida: [2,0,1]
```

1. (3 puntos) Leer y guardar en archivos según el ejemplo e insertar en una lista enlazada.
2. (3 puntos) Método para girar según el caso, recuerde que la menor complejidad es importante.
3. (3 puntos) Implementar la visualización de los datos usando recursividad.
4. (1 puntos) Desarrollar un caso de prueba(main).