

# CP05-May the for be with you

Curso 2025-2026



## 1. Divisiones

Dada una lista  $A$  y una lista  $B$  de enteros. Escriba cada par de enteros de  $B$  que es divisible por un entero de  $A$ .

## 2. Histograma

Se tiene una lista de intervalos  $[a_i, b_i)$  donde se cumple que  $b_i = a_{i+1}$  en la forma de tuplas  $(a_i, b_i)$ . Se tiene también una lista  $M$  de números. Construye una lista  $H$  que cumpla que en la posición  $i$ -ésima se guarda la cantidad de valores en  $M$  que cumplen que son mayores o iguales que  $a_i$  y menores que  $b_i$ .

## 3. Bounds I

Dada una lista de  $n$  enteros ordenados, potencialmente repetidos, y un entero  $v$ , obtenga el índice de la primera y de la última aparición de este valor en la lista.

**Nota:** Para hacerlo de forma eficiente utilice el enfoque de búsqueda binaria

## 4. Cartógrafo I

Se desean tomar anotaciones de las exploraciones en un archipiélago desconocido. Cada una de las exploraciones  $i$  sale a visitar una ubicación específica en las coordenadas  $x_i, y_i$ . Al terminar la expedición se informa si se encontró que en esta ubicación hay tierra o si se encontró agua. Si sabe que las expediciones recorrerán un territorio de tamaño  $n \times m$  y que al finalizar estas se habrán explorado cada una de las ubicaciones posibles entonces se puede construir un mapa del archipiélago.

Escriba un código que permita recibir como entrada la ubicación y resultado de cada una de las expediciones y luego se almacene el mapa de alguna forma para consultarlo.

## 5. Suspendiendo Lógica

Se tiene una lista con cada uno de los pares  $\langle x, y \rangle$  de una relación como tuplas. Dada la lista responde si:

1. La relación es reflexiva.
2. La relación es simétrica.
3. La relación es transitiva.

## 6. Bounds II

Dada una lista  $A$  de números ya ordenada, escriba un algoritmo que permita añadir un valor nuevo a esta lista manteniendo el orden.

**Nota:** No debe utilizar la función `sort` pero puede utilizarse `insert`.

## 7. Hallando Raíz

Dado un entero  $n$  calcule su raíz cuadrada aproximada. Sin usar **ninguna** función existente en python. **Ninguna**. Tampoco utilizar el operador `**`.

**Nota:** Recuerda que  $\sqrt{n}$  cumple que  $(\sqrt{n})^2 = n$ . También se cumple que  $\forall x < \sqrt{n} \mid x^2 < n$  y de igual manera  $\forall x > \sqrt{n} \mid x^2 > n$ .

## 8. La Criba

Si se desea conocer si **un** número es primo se puede comprobar su divisibilidad de la manera que se ha implementado en semanas anteriores, pero si tenemos **varios números** que debemos comprobar si son primos ese algoritmo se vuelve lento. Para esto se precálculan los primos desde 1 hasta un valor suficientemente grande utilizando la llamada *Criba de Eratóstenes* que consiste en construir una lista que cumpla que en la posición  $i$  se guarde si el número  $i$  es primo o no.

La construcción de esta lista se basa en que si un número  $p$  es primo entonces sus múltiplos, de la forma  $kp$  con  $k \in \mathbb{N}; k > 0$  no lo son. De igual forma si un número  $\alpha$  no es primo entonces existe un primo  $p_0 < \alpha$  que lo divide.

1. A partir de estas propiedades realiza la criba de Eratóstenes para todos los números naturales menores que  $10^6$ .

2. Si en lugar de guardar si un número es primo o no guardáramos todos los primos que lo dividen a él entonces podríamos descomponer un número como la multiplicación de sus factores primos. Implementa la variación de la criba que permite esto.

## 9. Rectángulo Máximo

Dada una matriz formada únicamente por 0s y 1s, encuentra el área del mayor rectángulo o submatriz que se pueda extraer de esta tal que todos los valores dentro de él sean 1s.

Ejemplo:

```
matrix = [  
    [0, 0, 1, 0, 1, 0, 0],  
    [1, 0, 1, 1, 1, 1, 1],  
    [0, 1, 0, 1, 1, 1, 0],  
    [1, 0, 0, 1, 1, 1, 0],  
    # El rectángulo mayor es:  
    # [., ., ., ., ., ., .]  
    # [., ., ., 1, 1, 1, .]  
    # [., ., ., 1, 1, 1, .]  
    # [., ., ., 1, 1, 1, .]  
    # [., ., ., ., ., ., .]  
    # Por tanto la respuesta es 9 (3*3)  
]
```

## 10. Buscaminas ahora 2D

Se tiene una cuadrícula/matriz de valores  $-1$  o  $0$ . Las casillas con valor  $-1$  contienen una bomba mientras que las que tienen valor  $0$  no. Tu tarea es cambiar los valores que almacena la matriz para que en una casilla se guarde  $-1$  si tiene una bomba,  $0$  si no tiene una bomba ni tampoco ninguna de sus casillas alrededor y un valor entre  $1$  y  $8$  que indique la cantidad de casillas adyacentes con bombas.

## 11. Conexiones

Se dice que cada par de personas en el mundo se encuentra a  $6$  o menos conexiones sociales de separación. Esto se basa en que la red de las conexiones humanas son de un tipo llamado “*small world*”. En este ejercicio tienes una lista de tamaño  $n$  donde cada índice representa a una persona. En cada posición se almacena a sus vez una lista de todas las personas que son amigos de la persona representada en esa posición. Por ejemplo, en el índice  $[1]$  se pudiera tener la lista  $[0, 5, 6]$ , esto me indicaría que  $1$  es amigo de  $0, 5$  y  $6$ .

Tu tarea es obtener cada par de personas a través de las cuales se puede establecer una o más conexiones de  $5$  personas distintas entre ellas y por tanto tienen  $6$  grados de separación o menos la una de la otra.

## 12. El Juego de la Vida

El “Juego de la Vida” es un autómata celular diseñado por John Conway. Se desarrolla en una cuadrícula de celdas, cada una de las cuales puede estar “viva” o “muerta”. La evolución de la cuadrícula se da en pasos o “generaciones” siguiendo unas reglas simples:

1. Una celda viva con menos de dos vecinos vivos, muere (por soledad).
2. Una celda viva con dos o tres vecinos vivos, sobrevive a la siguiente generación.
3. Una celda viva con más de tres vecinos vivos, muere (por sobrepoblación).
4. Una celda muerta con exactamente tres vecinos vivos, se convierte en una celda viva (por reproducción).

Implemente esta simulación.

1. Define una cuadrícula inicial (una lista de listas con 1 para vivo, 0 para muerto).
2. En un bucle principal, calcula el estado de la cuadrícula para la siguiente generación basándote en las reglas.
3. Imprime la cuadrícula en cada generación. El bucle debe ejecutarse un número determinado de veces o hasta que el usuario decida parar.