



Ejercicio 1. Dada una matriz determinar si se trata de una matriz triangular. Es decir, todos los elementos por debajo de la diagonal son cero.

Ejercicio 2. Dadas dos matrices A y B se desea multiplicarlas siguiendo esta especificación:

```
proc multiplicar (in m1: seq<seq<Z>>, in m2: seq<seq<Z>>, out res: seq<seq<Z>>) {
  Pre {completar}
  Post { |res| = |m1|  $\wedge_L$  ( $\forall i : \mathbb{Z}$ ) ( $0 \leq i < |m1| \rightarrow_L (res[i] = |m2[0]| \wedge_L (\forall j : \mathbb{Z}) (0 \leq j < |m2[0]| \rightarrow_L res[i][j] =$ 
     $\sum_{k=0}^{|m2|-1} m1[i][k] * m2[k][j]))$ ) }
}
```

a) completar la precondition del problema.

b) escribir un algoritmo que retorne AB

Ejercicio 3. Dada una matriz cuadrada triangular (es decir, todos los elementos por debajo de la diagonal son cero) de números enteros, escribir un algoritmo que calcule su determinante. Recordar que el determinante de una matriz triangular es el producto de los elementos de su diagonal.

Ejercicio 4. ★

a) Dada una matriz A , y un entero i escribir una función que retorne el vector resultante de multiplicar la fila i por cada una de las filas de la matriz. Escribir una versión iterativa y una versión recursiva de la misma.

b) Dada una matriz A , escribir una función que retorne $B = AA^t$ siendo A^t la matriz transpuesta de A . Escribir una versión iterativa y una versión recursiva de la misma.

Ejercicio 5. Escribir un algoritmo que calcule A^n siendo A una matriz cuadrada y n un número natural.

a) ¿Cuántas operaciones realiza esta función?

b) ¿Es posible disminuir la cantidad de operaciones drásticamente? Reescribir el algoritmo para aprovechar cuentas ya realizadas.

c) Escribir versiones recursivas e iterativas de los items anteriores.

Ejercicio 6. Escribir un algoritmo que dados una matriz A de $N \times N$, y números n_2 y m_2 devuelva la matriz resultante de re-dimensionar la matriz A con la nueva dimensión $n_2 \times m_2$. ¿Cuál es la precondition del problema?

Ejemplo, suponiendo $n_2 = 6$ y $m_2 = 2$:

A =

| | | | |
|---|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |

A' =

| | | | | | |
|---|---|---|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 |

Ejercicio 7. ★ Una matriz dispersa (o rala) es una matriz de gran tamaño en la que la mayor parte de sus elementos son cero. En este caso, muchas veces es conveniente cambiar la forma en que representamos estas matrices.

a) Especificar y escribir una función `vector<tuple<int, int, int>> aTriplas(vector<vector<int>> m)` que: dada una matriz A formada por vectores vectores, devuelva un vector de triplas (i, j, v) que contenga las posiciones de la matriz cuyos elementos sean distintos a cero junto a su valor.

- b) Especificar y escribir una función `vector<vector<int>> aTriplas(vector<tuple<int, int, int>> m)` que, dada una matriz representada como vector de triplas, genere la matriz asociada representada como vectores de vectores.

Ejercicio 8. Dada un arreglo multidimensional (tensor) de dimensiones $(N \times M \times T)$ que representa la medición de temperatura en distintos puntos de una zona en el tiempo (N y M dimensiones de una grilla que cubren el terreno, y T el tiempo), se pide:

- a) Calcular la temperatura promedio de cada punto de la zona (el resultado debe ser una matriz de dimensión $N \times M$).
- b) Calcular el promedio en la zona de la temperatura en cada momento (el resultado debe ser un vector de dimensión T).

Ejercicio 9. ★ *Búsqueda en Matrices:* Dada una matriz cuadrada de enteros y un entero x ,

1. Especificar el problema de retornar la fila y columna que contiene al elemento x (o $\langle -1, -1 \rangle$ si no existe ese elemento)
2. Escribir un algoritmo que cumpla con especificación del problema.
3. ¿Qué cantidad de iteraciones realiza el algoritmo en **peor caso**?

Ejercicio 10. Dada una matriz de números enteros con las filas ordenadas de menor a mayor y las columnas ordenadas de menor a mayor. Escribir un algoritmo que cuente la cantidad de veces que aparece un número en la matriz:

1. Suponiendo que se sabe no hay elementos repetidos.
2. Si se sabe que puede haber repetidos en las columnas (pero no en las filas).
3. Si tanto las filas como las columnas pueden tener elementos repetidos.

Ejercicio 11. Dada una matriz de booleanos de n filas y m columnas con n impar. Se sabe que hay exactamente una fila que no está repetida, y el resto se encuentra exactamente dos veces en la matriz.

- a) Escribir un algoritmo que devuelva un vector con los valores de la fila que no se repite.
- b) ¿Es posible un algoritmo que pase por cada casillero de la matriz sólo una vez? En caso de no haberlo resuelto con esta restricción, modificar el algoritmo para que la cumpla.
- c) La solución al ítem anterior, ¿utiliza vectores auxiliares?, en caso que lo haga, escribir una solución que no los necesite.

Ejercicio 12. ★ *Valles:* Dada una matriz cuadrada de enteros sin repetidos que representa la elevación de un terreno, un valle es una celda tal que ninguna de las celdas adyacentes tiene un valor menor (mínimo local).

Especificar y escribir un algoritmo ...

1. ... **iterativo** que busque algún valle a partir de la celda $\langle i, j \rangle$ de la matriz.
2. ... **recursivo** que encuentre la posición del valle más profundo alcanzable desde un punto dado $\langle i, j \rangle$.
3. ... **recursivo** que encuentre todos los caminos que conducen a valles desde un punto dado $\langle i, j \rangle$.
4. ... **iterativo** que resuelva los ítems anteriores (OPCIONAL).

Ejercicio 13. ★ *Wall Follower:* Dada una matriz cuadrada de enteros que representa un laberinto tal que:

- el valor de celda 0 indica que en esa celda no hay pared,
- el 1 indica que esa celda está ocupada por una pared, y
- el valor -1 indica que esa celda es la salida del laberinto.

Suponiendo que todos los muros del laberinto están conectados entre sí, escribir el algoritmo que a partir de la entrada del laberinto (la celda $\langle 0, 0 \rangle$) busca el camino para salir del laberinto, retornando la ubicación de la salida y la cantidad de celdas que debió recorrer. **Importante:** El algoritmo no puede *atravesar* los muros.