P10: Matrices

Algoritmos y Estructuras de Datos I

Departamento de Computación
Universidad de Buenos Aires

Primer Cuatrimestre 2020

Introducción

- *Wikipedia dixit*: Una matriz es un *arreglo bidimensional* de números (llamados entradas de la matriz) ordenados en filas (o renglones) y columnas, donde una fila es cada una de las líneas horizontales de la matriz y una columna es...

Introducción

- *Wikipedia dixit*: Una matriz es un *arreglo bidimensional* de números (llamados entradas de la matriz) ordenados en filas (o renglones) y columnas, donde una fila es cada una de las líneas horizontales de la matriz y una columna es...
- Una forma de verlo es como una lista cuyos elementos están definidos por dos índices. El primero representa una fila y el segundo una columna.

Introducción

- *Wikipedia dixit*: Una matriz es un *arreglo bidimensional* de números (llamados entradas de la matriz) ordenados en filas (o renglones) y columnas, donde una fila es cada una de las líneas horizontales de la matriz y una columna es...
- Una forma de verlo es como una lista cuyos elementos están definidos por dos índices. El primero representa una fila y el segundo una columna.
- O bien, como una tabla de entradas cuyos elementos se acceden con un índice para la fila y otro para la columna.

Repaso: Matrices en C++

Las representamos como vectores de vectores de **int**, ejemplo:
`vector<vector<int> > M;`
Asumiendo M representa una matriz válida de al menos 1x1, la cantidad de filas está dada por `M.size()` y la cantidad de columnas por `M[0].size()`.
Y, para acceder (o asignar) al elemento en la fila i y columna j, usamos:
`M[i][j]`
Finalmente, podemos copiar matrices asignándolas, y también podemos construirlas fila a fila (y construir cada fila columna a columna) con el operador `push_back`.

Ejercicio 1: Rotación

Dada la siguiente especificación, escribir un programa que cumpla con lo pedido.

proc rotación (in m: $seq\langle seq\langle \mathbb{Z} \rangle\rangle$, in f: $\mathbb{Z}$, in c: $\mathbb{Z}$, out res: $seq\langle seq\langle \mathbb{Z} \rangle\rangle$) {

Pre $\{(\forall i : \mathbb{Z})(0 \leq i < |m| \longrightarrow_L |m[i]| = |m[0]|)\}$

Post $\{|res| = |m| \wedge_L (\forall i : \mathbb{Z})(0 \leq i < |m| \longrightarrow_L (|res[i]| = |m[i]| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |m[i]| \longrightarrow_L res[i][j] = m[(i + f) mod |m|][(j + c) mod |m|])))\}$

}

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.
- Sea M una matriz que representa un terreno montañoso. Cada celda contendrá un valor distinto de altitud tomado a partir del terreno.
  - Se desea encontrar algún valle, que sea alcanzable desde el casillero (i,j), y al que se pueda llegar por un camino que solamente descienda en altitud. Se considera un valle a cualquier mínimo local de altitud.

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.
- Sea M una matriz que representa un terreno montañoso. Cada celda contendrá un valor distinto de altitud tomado a partir del terreno.
  - Se desea encontrar algún valle, que sea alcanzable desde el casillero (i,j), y al que se pueda llegar por un camino que solamente descienda en altitud. Se considera un valle a cualquier mínimo local de altitud.
  - Ahora, entre todos los valles, se quiere el más bajo al que se pueda llegar desde ese punto.

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.
- Sea M una matriz que representa un terreno montañoso. Cada celda contendrá un valor distinto de altitud tomado a partir del terreno.
  - Se desea encontrar algún valle, que sea alcanzable desde el casillero (i,j), y al que se pueda llegar por un camino que solamente descienda en altitud. Se considera un valle a cualquier mínimo local de altitud.
  - Ahora, entre todos los valles, se quiere el más bajo al que se pueda llegar desde ese punto.
    TIP: primero calcular todos los casilleros alcanzables por caminos descendientes a partir de un punto y luego evaluar cuál es el mínimo.

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.
- Sea M una matriz que representa un terreno montañoso. Cada celda contendrá un valor distinto de altitud tomado a partir del terreno.
  - Se desea encontrar algún valle, que sea alcanzable desde el casillero (i,j), y al que se pueda llegar por un camino que solamente descienda en altitud. Se considera un valle a cualquier mínimo local de altitud.
  - Ahora, entre todos los valles, se quiere el más bajo al que se pueda llegar desde ese punto.
    TIP: primero calcular todos los casilleros alcanzables por caminos descendientes a partir de un punto y luego evaluar cuál es el mínimo.
    TIP del **TIP**: ¿Será más sencillo utilizar recursión para este problema?

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.
- Sea M una matriz que representa un terreno montañoso. Cada celda contendrá un valor distinto de altitud tomado a partir del terreno.
  - Se desea encontrar algún valle, que sea alcanzable desde el casillero (i,j), y al que se pueda llegar por un camino que solamente descienda en altitud. Se considera un valle a cualquier mínimo local de altitud.
  - Ahora, entre todos los valles, se quiere el más bajo al que se pueda llegar desde ese punto.
    TIP: primero calcular todos los casilleros alcanzables por caminos descendientes a partir de un punto y luego evaluar cuál es el mínimo.
    TIP del **TIP**: ¿Será más sencillo utilizar recursión para este problema?
  - Implementar versiones iterativas y recursivas.

Ejercicio 2: Terrenos

- Dada una matriz de n filas y m columnas, se quiere obtener la lista de casilleros vecinos de un casillero (i,j) dado.
- Sea M una matriz que representa un terreno montañoso. Cada celda contendrá un valor distinto de altitud tomado a partir del terreno.
  - Se desea encontrar algún valle, que sea alcanzable desde el casillero (i,j), y al que se pueda llegar por un camino que solamente descienda en altitud. Se considera un valle a cualquier mínimo local de altitud.
  - Ahora, entre todos los valles, se quiere el más bajo al que se pueda llegar desde ese punto.
    TIP: primero calcular todos los casilleros alcanzables por caminos descendientes a partir de un punto y luego evaluar cuál es el mínimo.
    TIP del **TIP**: ¿Será más sencillo utilizar recursión para este problema?
  - Implementar versiones iterativas y recursivas.
  - **Bonus:** supongamos que ahora se puede avanzar a través de los bordes y salir por el otro lado (como en la viborita). ¿Qué modificaciones se deben hacer?

Ejercicio 3: Ejercicio de Parcial

Una matriz se dice de *filas batidas* si todas las filas (i) contienen los mismos valores aunque no necesariamente en el mismo orden y (ii) todos los valores aparecen la misma cantidad de veces. La primera de las matrice sque se exhiben a continuación es de *filas batidas* y las restantes dos no lo son. Se pide escribir un programa en C++ que dada una matriz de $n \times m$ determine si es o no de *filas batidas*.

$$A = \begin{bmatrix} 1 & 2 & 6 \\ 6 & 1 & 2 \\ 2 & 6 & 1 \\ 1 & 2 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 3 \\ 3 & 1 & 1 \\ 3 & 3 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix}$$