CS2023 - Aula de Ejercicios Nº 13

Brenner H. Ojeda Rios

Semestre 2024-1

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

Ejercicios

1. (6 pts) Usar BFS

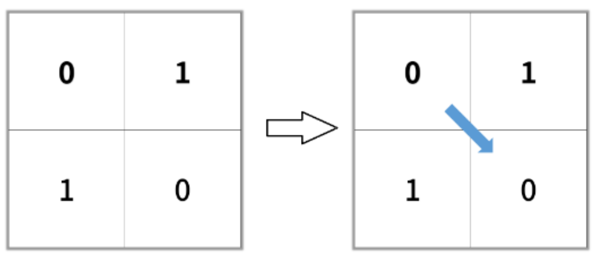
   Dada una de matriz binaria $n \times n$, devuelve la longitud del *camino claro* más corto en la matriz. Si no hay un camino claro, devuelve -1.

   Una camino claro en una matriz binaria es una ruta desde la celda superior izquierda (es decir, (0, 0)) hasta la celda inferior derecha (es decir, (n - 1, n - 1)) tal que:

   - Todas las celdas visitadas del camino son 0.
   - Todas las celdas adyacentes del camino están conectadas en 8 direcciones (es decir, son diferentes y comparten un borde o una esquina).

   La longitud de un *camino claro* es el número de celdas visitadas de este camino.

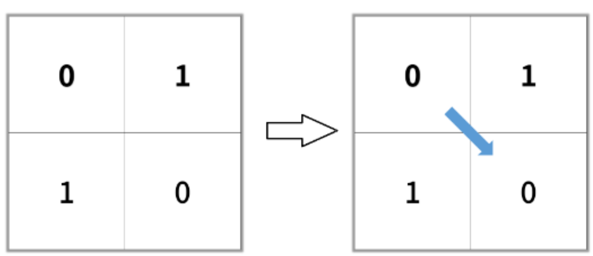
   Ejemplo 1:

   `Input: grid = [[0,1],[1,0]]`

   `Output: 2`

   Ejemplo 2:

   `Input: grid = [[0,0,0],[1,1,0],[1,1,0]]`

   `Output: 4`

   Ejemplo 3:

   `Input: grid = [[1,0,0],[1,1,0],[1,1,0]]`

   `Output: -1`

Restricciones:

- $n = grid.length$
- $n = grid[i].length$
- $1 \leq n \leq 100$
- $grid[i][j] es 0 o 1$

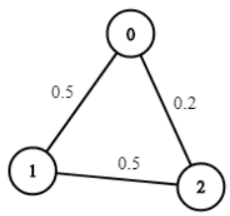
2. (7 pts) Usar Bellman Ford

Se le proporciona un grafo ponderado no dirigido de n nodos (con índice 0), representado por una lista de aristas donde `edge[i]` $= [a, b]$ es una arista no dirigida que conecta los nodos a y b con una probabilidad de éxito al atravesarlo de `succProb[i]`.

Dado dos nodos `start` y `end`, encuentre el camino con la máxima probabilidad de éxito para ir de `start` a `end` y devuelva su probabilidad de éxito.

Si no hay una ruta de `start` a `end`, devuelve 0. Su respuesta será aceptada si difiere de la respuesta correcta en como máximo 1e-5. Hay como máximo un arista entre cada dos nodos.

- **Ejemplo 1:**

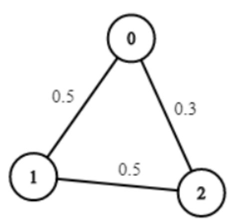


`Input: n = 3, edges = [[0,1],[1,2],[0,2]], succProb = [0.5,0.5,0.2], start = 0, end = 2`

`Output: 0.25000`

`Explicación:` Hay dos caminos de principio a fin, uno tiene una probabilidad de éxito $= 0{,}2$ y el otro tiene $0{,}5 * 0{,}5 = 0{,}25$.

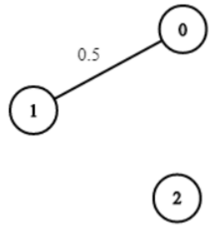
- **Ejemplo 2:**



`Input: n = 3, edges = [[0,1],[1,2],[0,2]], succProb = [0.5,0.5,0.3], start = 0, end = 2`

`Output: 0.30000`

- **Ejemplo 3:**



`Input: n = 3, edges = [[0,1]], succProb = [0.5], start = 0, end = 2`
`Output: 0.00000`
`Explicación:` No hay un camino de 0 a 2.

Restricciones:

- $1 \leq n \leq 10^4$
- $0 \leq$ `start, end` $< n$
- `start` $\neq$ `end`
- $0 \leq$ `a, b` $< n$
- $0 \leq$ `len(succProb)` $=$ `len(edges)` $\leq 2 \times 10^4$
- $0 \leq$ `succProb[i]` ≤ 1

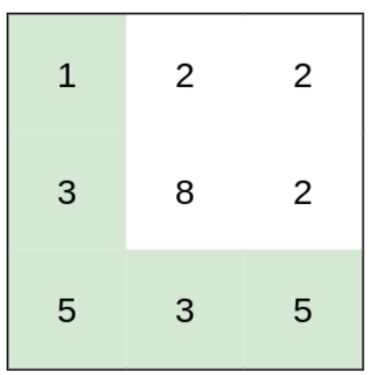
3. (7 pts) Usar Dijkstra

Eres un excursionista que se prepara para una próxima caminata. Se le dan una matriz de alturas, de tamaño `rows x columns`, donde `alturas[fil][col]` representa la altura de la celda (fila, columna). Está situado en la celda superior izquierda (0, 0) y espera viajar a la celda inferior derecha (filas-1, columnas-1) (es decir, indexado en 0). Puede moverte hacia arriba, abajo, izquierda o derecha y deseas encontrar una ruta que requiera el mínimo esfuerzo.

El esfuerzo de una ruta es la máxima diferencia absoluta de cotas entre dos celdas consecutivas de la ruta.

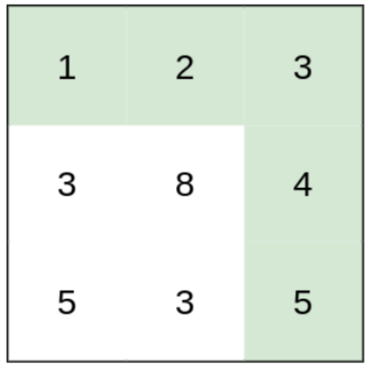
Devuelve el esfuerzo mínimo necesario para viajar desde la celda superior izquierda a la celda inferior derecha.

- **Ejemplo 1:**



`Input: heights = [[1,2,2],[3,8,2],[5,3,5]]`
`Output: 2`
`Explicación:` La ruta de [1,3,5,3,5] tiene una diferencia absoluta máxima de 2 en celdas consecutivas. Esto es mejor que la ruta de [1,2,2,2,5], donde la diferencia absoluta máxima es 3.

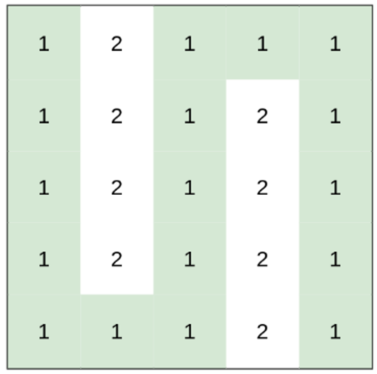
- **Ejemplo 2:**



`Input: heights = [[1,2,3],[3,8,4],[5,3,5]]`

`Output: 1` (La ruta de [1,2,3,4,5] tiene una diferencia absoluta máxima de 1 en celdas consecutivas, que es mejor que la ruta [1,3,5,3,5].)

- **Ejemplo 3:**



`Input: heights = [[1,2,1,1,1],[1,2,1,2,1],[1,2,1,2,1],[1,2,1,2,1],[1,1,1,2,1]]`
`Output: 0` (Esta ruta no requiere ningún esfuerzo.)

Restricciones:

- `rows` $=$ `heights.length`
- `columns` $=$ `heights[i].length`
- $1 \leq$ `rows, columns` ≤ 100
- $1 \leq$ `heights[i][j]` $\leq 10^6$