Prueba de escritorio

Dados los siguientes nodos y aristas: Nodo 1 a 2 con peso 2. Nodo 2 a 4 con peso 4. Nodo 1 a 4 con peso 5. Nodo 4 a 3 con peso 2.

Prueba de escritorio

Paso 1: Inicialización de la matriz de distancias La matriz de adyacencia inicial será la siguiente, donde INF indica que no hay un camino directo entre dos nodos:

x 1 2 3 4
1 0 2 INF 5
2 INF 0 INF 4
3 INF INF 0 INF
4 INF INF 2 0

Prueba de escritorio

Explicación de la matriz:
- Nodo 1 a 1: Distancia 0, ya que es el mismo nodo.
- Nodo 1 a 2: Distancia 2 (directo, como dado).
- Nodo 1 a 3: No hay camino directo, así que INF.
- Nodo 1 a 4: Distancia 5 (directo, como dado).
- Nodo 2 a 1: No hay camino directo, así que INF.
- Nodo 2 a 2: Distancia 0, ya que es el mismo nodo.
- Nodo 2 a 3: No hay camino directo, así que INF.
- Nodo 2 a 4: Distancia 4 (directo, como dado).
- Nodo 3 a 1, 3 a 2, 3 a 4: No hay caminos directos, así que INF.
- Nodo 3 a 3: Distancia 0, ya que es el mismo nodo.
- Nodo 4 a 1, 4 a 2: No hay caminos directos, así que INF.
- Nodo 4 a 3: Distancia 2 (directo, como dado).
- Nodo 4 a 4: Distancia 0, ya que es el mismo nodo.

Prueba de escritorio

Ahora iteramos sobre cada nodo intermedio k (1, 2, 3 y 4) para actualizar las distancias mínimas entre todos los pares de nodos (i, j).

Iteración 1: k = 1
Para cada par de nodos (i, j), verificamos si pasar por el nodo intermedio 1 da una distancia más corta:
No se actualizan valores ya que no hay una mejora en las distancias.

Matriz actualizada:

x 1 2 3 4
1 0 2 INF 5
2 INF 0 INF 4

3 INF INF 0 INF
4 INF INF 2 0

Prueba de escritorio

Iteración 2: k = 2 Para i = 1, j = 4: dist[1][4] = min(dist[1][4], dist[1][2] + dist[2][4]) dist[1][4] = min(5, 2 + 4) = 5 (sin cambios) No hay más cambios en esta iteración.

Matriz actualizada:

x 1 2 3 4
1 0 2 INF 5
2 INF 0 INF 4
3 INF INF 0 INF
4 INF INF 2 0
Prueba de escritorio

Iteración 3: k = 3
No hay actualizaciones porque el nodo 3 no tiene conexiones que permitan mejorar otras distancias.

Matriz actualizada:

x 1 2 3 4
1 0 2 INF 5
2 INF 0 INF 4
3 INF INF 0 INF
4 INF INF 2 0

Prueba de escritorio

Iteración 4: k = 4 Para i = 1, j = 3: dist[1][3] = min(dist[1][3], dist[1][4] + dist[4][3]) dist[1][3] = min(INF, 5 + 2) = 7

Para i = 2, j = 3: dist[2][3] = min(dist[2][3], dist[2][4] + dist[4][3]) dist[2][3] = min(INF, 4 + 2) = 6

Matriz actualizada:

x 1 2 3 4
1 0 2 INF 5
2 INF 0 INF 4
3 INF INF 0 INF
4 INF INF 2 0

Prueba de escritorio

Iteración 4: k = 4 Para i = 1, j = 3: dist[1][3] = min(dist[1][3], dist[1][4] + dist[4][3]) dist[1][3] = min(INF, 5 + 2) = 7

Para i = 2, j = 3: dist[2][3] = min(dist[2][3], dist[2][4] + dist[4][3]) dist[2][3] = min(INF, 4 + 2) = 6

Matriz actualizada:

x	1	2	3	4
1	0	2	7	5
2	INF	0	6	4
3	INF	INF	0	INF
4	INF	INF	2	0

Resultado final

La matriz final de distancias mínimas entre todos los pares de nodos es:

x 1 2 3 4
1 0 2 7 5
2 INF 0 6 4
3 INF INF 0 INF
4 INF INF 2 0

Esta matriz representa las distancias más cortas entre todos los nodos tras aplicar el algoritmo de Floyd-Warshall.