Estructuras de Datos

Noviembre 6/2024

Laboratorio Grafos

John Corredor Franco

Daniel Rosas

- DOCUMENTACIÓN:

Algoritmo de Dijkstra: Explicación Paso a Paso

El algoritmo de Dijkstra es un algoritmo de búsqueda de caminos en grafos ponderados que encuentra la ruta de menor costo (distancia mínima) desde un nodo de origen a todos los otros nodos del grafo. Fue desarrollado por el científico holandés Edsger W. Dijkstra en 1956. Este algoritmo es útil para aplicaciones como sistemas de navegación, redes de comunicación y análisis de rutas.

Su objetivo es, encontrar el camino de menor costo entre un nodo inicial y los demás nodos de un grafo ponderado y dirigido o no dirigido.

- Grafo: Conjunto de nodos conectados por aristas, donde cada arista tiene un peso o costo asociado.
- Nodo (Vertice): Representa un punto en el grafo.
- Peso/Costo: Valor que representa la distancia o costo de recorrer una arista entre dos nodos.

Descripción del Algoritmo:

La Inicialización:

- Asigna una distancia inicial de 0 al nodo de inicio (nodo fuente) y distancia infinita (∞) a todos los otros nodos.

Marca todos los nodos como no visitados.

Bucle Principal:

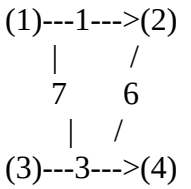
- Selecciona el nodo no visitado con la distancia mínima (al inicio, el nodo de inicio).
- Marca el nodo como visitado (ya no se considerará en las siguientes iteraciones).
- Para cada vecino de este nodo:
  - Calcula la distancia tentativa desde el nodo de inicio hasta el vecino.
  - Si esta distancia es menor que la distancia actualmente registrada para el vecino, actualiza la d distancia.
  -Guarda el camino previo para poder reconstruir la ruta

Finalización:

- El proceso continúa hasta que todos los nodos hayan sido visitados o no haya más nodos alcanzables.
- Los valores en el vector de distancias representan la distancia mínima desde el nodo de inicio hasta cada nodo.

Ejemplo Paso a Paso de lo realizado:

Imaginemos un grafo con nodos y pesos representados de la siguiente manera:



- Queremos encontrar la ruta más corta desde el nodo 1 a los demás nodos.
- Paso 1: Inicialización

Nodo	Distancia desde 1	Visitado
1	0	No
2	∞	No
3	∞	No
4	∞	No

Nodo de inicio: 1, con distancia 0.
Todos los demás nodos tienen distancia infinita (∞).

Paso 2: Bucle Principal

Iteración 1:

Seleccionamos el nodo 1 (distancia 0).
Visitamos sus vecinos:
Nodo 2: La distancia tentativa desde 1 es 1. Actualizamos la distancia de 2 a 1.
Nodo 3: La distancia tentativa desde 1 es 7. Actualizamos la distancia de 3 a 7.

Nodo	Distancia desde 1	Camino
1	0	1
2	1	1 → 2
3	7	1 → 3
4	∞	-

Iteración 2:

Seleccionamos el nodo 2 (distancia 1).
Visitamos su vecino:
Nodo 4: La distancia tentativa desde 2 es 6. Actualizamos la distancia de 4 a 6.

Nodo	Distancia desde 1	Camino
1	0	1
2	1	1 → 2
3	7	1 → 3
4	6	1 → 2 → 4

Iteración 3:

Seleccionamos el nodo 4 (distancia 6).

No hay vecinos no visitados.

Iteración 4:

Seleccionamos el nodo 3 (distancia 7).
No hay vecinos no visitados.

Resultado Final

Nodo	Distancia desde 1	Camino
1	0	1
2	1	1 → 2
3	7	1 → 3
4	6	1 → 2 → 4

Código Ejemplo en C++

- VER ANEXO CODIGO FUENTE

Consideraciones que debemos tener en cuenta.

Complejidad Temporal: O((N+ E) log N, donde V es el número de nodos y E el número de aristas.
Limitaciones: Este algoritmo solo funciona en grafos con pesos positivos.
Aplicaciones: Es utilizado en sistemas de navegación GPS, enrutamiento de redes, y planificación logística.