

INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA
DE INGENIERÍA CAMPUS ZACATECAS

REPORTE DE PRÁCTICA 03

ALUMNO: ALONSO ESTANISLAO GÓMEZ RODRÍGUEZ

GRUPO: 3CM2

DOCENTE: M. EN C. ERIKA SÁNCHEZ FEMAT

ANÁLISIS Y DISEÑO DE ALGORITMOS

diciembre, 2023

Introducción

El algoritmo de Dijkstra es una herramienta fundamental en el ámbito de la teoría de grafos y la optimización de rutas. Desarrollado por Edsger Dijkstra en 1956, este algoritmo resuelve el problema de encontrar los caminos más cortos desde un vértice de origen a todos los demás vértices en un grafo ponderado dirigido.

Se utiliza en una variedad de aplicaciones, como en la planificación de rutas en mapas, en redes de comunicación para encontrar la ruta más eficiente entre dos nodos, y en problemas logísticos donde se busca minimizar el costo de transporte.

Desarrollo

Un grafo ponderado dirigido es una estructura de datos que representa una red de conexiones entre elementos (nodos) con pesos asociados a las conexiones (aristas). En este tipo de grafos, cada arista tiene un peso positivo que representa la distancia, tiempo, costo o cualquier otra magnitud relevante entre los nodos conectados. El peso de un arista puede ser positivo, negativo o cero, dependiendo de si la conexión es deseable, indeseable o inalcanzable, respectivamente.

La problemática de encontrar la ruta más corta en un grafo se refiere a la búsqueda de un camino entre dos nodos o vértices en un grafo ponderado dirigido, de tal manera que la suma de los pesos de las aristas en el camino sea mínima. Este problema es conocido como el problema del camino más corto y es un problema clásico en la teoría de grafos y en la informática.

El algoritmo de Dijkstra es un enfoque eficiente para resolver este problema en grafos ponderados dirigidos. Este algoritmo, desarrollado por el matemático indio Dijkstra, trabaja en un vértice de inicio y calcula sucesivamente las distancias mínimas a todos los otros vértices en el grafo. La idea clave del algoritmo es utilizar una cola de prioridad para procesar los vértices de manera eficiente, asegurando que el vértice actualmente procesado siempre sea el vértice más prometedor.

El algoritmo de Dijkstra es un método eficiente para encontrar la ruta más corta entre un nodo de origen y todos los demás nodos en un grafo ponderado dirigido. Su funcionamiento básico consiste en iniciar en el nodo de origen y analizar el grafo para encontrar el camino más corto entre ese nodo y todos los otros nodos en el grafo. El algoritmo utiliza una cola de prioridad para seleccionar siempre el nodo con la distancia más corta pendiente de procesar, y recalcula la distancia a todos los vértices adyacentes al nodo actual. El algoritmo continúa este proceso hasta que se hayan procesado todos los nodos y se hayan encontrado las distancias más cortas a todos los demás nodos en el grafo.

Implementación del algoritmo:

- Definir una clase Graph que represente el grafo ponderado dirigido. Esta clase debe tener un constructor para inicializar la estructura del grafo, un método add vértice para agregar vértices y un método add arista para agregar aristas con pesos.
- Implementar la función dijkstra que tome el grafo y un vértice de inicio como entrada y devuelva los caminos mínimos desde el vértice de inicio a todos los demás vértices. Utilizar un diccionario para mantener un registro de las distancias mínimas.

Ejemplo de uso:

- Crear un grafo ponderado dirigido de ejemplo y aplicar el algoritmo de Dijkstra para encontrar los caminos mínimos desde un vértice de inicio.
- Mostrar los resultados obtenidos y explicar el significado de los caminos mínimos encontrados.

La complejidad del algoritmo de Dijkstra depende de la estructura de datos utilizada para implementar la cola de prioridad. En el peor de los casos, la complejidad del algoritmo de Dijkstra es de $O((E+V)\log V)$, donde V es el número de vértices y E es el número de aristas en el grafo. Esta complejidad se obtiene cuando se utiliza una cola de prioridad implementada con un heap binario, como en el caso de la implementación estándar del algoritmo.

Resultados

```
Camino s mínimos desde el vértice A:  
Vértice G: Distancia 16  
Tiempo de ejecución: 0.0 segundos  
Vértice A: Distancia 0  
Tiempo de ejecución: 0.0 segundos  
Vértice C: Distancia 3  
Tiempo de ejecución: 0.0 segundos  
Vértice E: Distancia 11  
Tiempo de ejecución: 0.0 segundos  
Vértice B: Distancia 1  
Tiempo de ejecución: 0.0 segundos  
Vértice F: Distancia 14  
Tiempo de ejecución: 0.0 segundos  
Vértice D: Distancia 4  
Tiempo de ejecución: 0.0 segundos
```

```
Camino m nimos desde el v rtice A:  
V rtice D: Distancia 8  
Tiempo de ejecuci n: 0.0 segundos  
V rtice G: Distancia 20  
Tiempo de ejecuci n: 0.0 segundos  
V rtice B: Distancia 5  
Tiempo de ejecuci n: 0.0 segundos  
V rtice F: Distancia 18  
Tiempo de ejecuci n: 0.0 segundos  
V rtice C: Distancia 4  
Tiempo de ejecuci n: 0.0 segundos  
V rtice A: Distancia 0  
Tiempo de ejecuci n: 0.0 segundos  
V rtice E: Distancia 15  
Tiempo de ejecuci n: 0.0 segundos
```

```
Camino m nimos desde el v rtice A:  
V rtice D: Distancia 83  
Tiempo de ejecuci n: 0.0 segundos  
V rtice A: Distancia 0  
Tiempo de ejecuci n: 0.0 segundos  
V rtice G: Distancia 115  
Tiempo de ejecuci n: 0.0 segundos  
V rtice E: Distancia 100  
Tiempo de ejecuci n: 0.0 segundos  
V rtice F: Distancia 105  
Tiempo de ejecuci n: 0.0 segundos  
V rtice B: Distancia 58  
Tiempo de ejecuci n: 0.0 segundos  
V rtice C: Distancia 54  
Tiempo de ejecuci n: 0.0 segundos
```

```
Camino s m nimos desde el v rtice A:  
V rtice F: Distancia 79  
Tiempo de ejecuci n: 0.0 segundos  
V rtice D: Distancia 6  
Tiempo de ejecuci n: 0.0 segundos  
V rtice C: Distancia 4  
Tiempo de ejecuci n: 0.0 segundos  
V rtice E: Distancia 23  
Tiempo de ejecuci n: 0.0 segundos  
V rtice B: Distancia 8  
Tiempo de ejecuci n: 0.0 segundos  
V rtice A: Distancia 0  
Tiempo de ejecuci n: 0.0 segundos  
V rtice G: Distancia 89  
Tiempo de ejecuci n: 0.0 segundos
```

```
Camino s m nimos desde el v rtice A:  
V rtice F: Distancia 919  
Tiempo de ejecuci n: 0.0 segundos  
V rtice A: Distancia 0  
Tiempo de ejecuci n: 0.0 segundos  
V rtice B: Distancia 108  
Tiempo de ejecuci n: 0.0 segundos  
V rtice D: Distancia 145  
Tiempo de ejecuci n: 0.0 segundos  
V rtice E: Distancia 323  
Tiempo de ejecuci n: 0.0 segundos  
V rtice C: Distancia 172  
Tiempo de ejecuci n: 0.0 segundos  
V rtice G: Distancia 929  
Tiempo de ejecuci n: 0.0 segundos
```