



INSTITUTO POLITÉCNICO NACIONAL
Unidad Profesional Interdisciplinaria de Ingeniería
Campus Zacatecas.

MATERIA:
Análisis y Diseño de Algoritmos

DOCENTE:
Erika Sanchez Femat

PRACTICA 3
o Dijkstra

ALUMNO:

- o Fernando Antonio García Ruiz

INTRODUCCIÓN

El algoritmo de Dijkstra, desarrollado por el renombrado científico computacional Edsger Dijkstra en 1956, ha dejado una marca indeleble en el campo de la informática y la teoría de grafos. Este algoritmo, concebido en los albores de la era de las computadoras, se ha convertido en una herramienta fundamental para encontrar el camino más corto entre dos puntos en un grafo ponderado.

Edsger Dijkstra ideó este algoritmo con el objetivo de resolver problemas de optimización en el ámbito de las redes de transporte y telecomunicaciones. En sus primeros años, la informática enfrentó el desafío de optimizar las rutas de manera eficiente, y el algoritmo de Dijkstra surgió como una respuesta ingeniosa a esta necesidad. A lo largo de las décadas, el algoritmo ha encontrado aplicaciones en diversas disciplinas, desde el diseño de circuitos hasta la planificación logística.

¿COMO FUNCIONA?

El algoritmo de Dijkstra opera en grafos ponderados, donde cada arista tiene asignado un peso no negativo. Su función principal es encontrar el camino más corto desde un nodo de origen hasta todos los demás nodos en el grafo. El proceso comienza asignando una distancia inicial de cero al nodo de origen y considerando todas las demás distancias como infinito.

A través de iteraciones, el algoritmo evalúa y actualiza las distancias tentativas hacia los nodos vecinos, seleccionando en cada paso el nodo con la menor distancia tentativa. A medida que se visitan los nodos, el algoritmo acumula información sobre las distancias más cortas hasta esos puntos. Este proceso se repite hasta que se han explorado todos los nodos o se alcanza el nodo de destino.

El algoritmo de Dijkstra destaca por su eficiencia y simplicidad conceptual. Sin embargo, es crucial destacar su limitación: no maneja adecuadamente pesos de arista negativos, ya que podría conducir a resultados incorrectos. Para enfrentar este problema, han surgido variantes del algoritmo, pero la versión clásica de Dijkstra sigue siendo un pilar en la resolución de problemas de optimización de rutas.

En conclusión, el algoritmo de Dijkstra es un hito en la historia de la informática, un instrumento esencial para encontrar caminos más cortos en grafos ponderados y un testimonio del ingenio de Edsger Dijkstra en la búsqueda de soluciones eficientes y elegantes en el mundo de los algoritmos y la computación.

EVIDENCIAS

The image shows two identical terminal windows side-by-side, both running a Python script to find shortest paths in a graph. The script uses Dijkstra's algorithm to calculate the minimum distance from node A to nodes B and C. The output for each run is as follows:

```
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Caminos mínimos desde A: {'A': 0, 'B': 3, 'C': 5}  
Tiempo de ejecución: 0.0 segundos  
---  
Grafo:  
A: {'B': 3, 'C': 5}  
B: {'C': 2}  
C: {}  
  
Tiempo promedio de ejecución: 0.0 segundos  
PS C:\Users\Javier Alejandro\OneDrive\Documentos\Programación\Programas\A y O_Algoritmos> █
```

CONCLUSIÓN

En conclusión, el código es relativamente claro y sigue la lógica del algoritmo de Dijkstra de manera comprensible. Las estructuras de datos utilizadas, como diccionarios y colas de prioridad, son apropiadas y facilitan la lectura del código.