ceti

CENTRO DE ENSEÑANZA
TÉCNICA INDUSTRIAL

INTELIGENCIA ARTIFICIAL

Saul Armando Cuenca Martínez
21110324

"ALGORITMO DIJKSTRA"

Parte Teórica

¿Qué es el algoritmo Dijkstra ?

Al algoritmo de Dijkstra se lo conoce también como algoritmo de caminos mínimos. Fue creado e ideado por el científico computacional holandés Edsger Wybe Dijkstra, en 1959. Este algoritmo es utilizado para determinar el camino más corto para ejecutar desde un vértice origen hasta el resto de los vértices ubicados en un grafo con pesos en cada arista.
Un grafo, por si no conoces el término, se refiere a un tipo de representación simbólica o gráfica de distintos tipos de elementos que conforman un conjunto o sistema.

Entonces, supongamos que de un vértice de origen tienes muchos caminos, sin saber cuál es el más corto. Este algoritmo te va a permitir recorrer todos los demás vértices y, cuando obtiene el más corto, lo ejecuta y se detiene.

¿Para qué sirve el algoritmo Dijkstra ?

El algoritmo de Dijkstra puede tener diversas utilidades cuando es aplicado dentro de los grafos. Puede ser utilizado para:

- La distribución de los productos a una o varias redes de establecimientos comerciales.
- Para los servicios de distribución de los correos postales.
- Para un grafo dirigido ponderado (G= (V, A)).

El problema del camino más corto de un vértice a otro consiste en determinar el camino de menor costo, desde un vértice u a otro vértice v. El costo de un camino es la suma de los costos (pesos) de los arcos que lo conforman.

¿Cómo se implementa el algoritmo Dijkstra en el mundo?

El algoritmo de Dijkstra se implementa en el mundo en una amplia variedad de aplicaciones y sistemas en los que se requiere encontrar rutas o caminos más cortos en redes, como redes de carreteras, redes de transporte público, redes de comunicaciones, sistemas de navegación, sistemas de logística, juegos, y más. A continuación, se presentan algunas de las formas en que el algoritmo de Dijkstra se implementa en el mundo real:

Sistemas de Navegación GPS: Los dispositivos de navegación GPS utilizan el algoritmo de Dijkstra para encontrar la ruta más corta desde un punto de inicio a un destino. Esto es fundamental en aplicaciones de navegación como Google Maps, Waze y sistemas de navegación incorporados en automóviles.

Redes de Transporte Público: Las aplicaciones de transporte público utilizan el algoritmo de Dijkstra para planificar rutas óptimas en sistemas de transporte público, incluyendo autobuses, trenes, tranvías y metros. Ayuda a los pasajeros a encontrar las rutas más eficientes para llegar a su destino.

Redes de Telecomunicaciones: Las empresas de telecomunicaciones utilizan Dijkstra para enrutar llamadas y datos a través de sus redes, asegurando que la información se transmita de la manera más eficiente posible.

Optimización de Rutas en Logística: Las empresas de logística y transporte utilizan el algoritmo de Dijkstra para planificar rutas de entrega óptimas para camiones y vehículos, minimizando los costos y el tiempo de viaje.

Redes Sociales y Grafos: En redes sociales y aplicaciones de grafos, como Facebook, LinkedIn y Twitter, el algoritmo de Dijkstra puede usarse para encontrar conexiones entre usuarios o para recomendar conexiones basadas en amigos mutuos u otros factores.

Juegos y Simulaciones: Los desarrolladores de videojuegos y simulaciones utilizan Dijkstra para simular el comportamiento de personajes o agentes en el juego y encontrar caminos óptimos en el entorno virtual.

Optimización de Rutas Aéreas: En la aviación, se utiliza el algoritmo de Dijkstra para calcular rutas de vuelo óptimas, teniendo en cuenta factores como el consumo de combustible y las condiciones meteorológicas.

Planificación de Redes de Fibra Óptica: En el campo de las telecomunicaciones, el algoritmo de Dijkstra es utilizado para planificar redes de fibra óptica, determinando la ubicación óptima de los cables para minimizar la longitud total y maximizar la eficiencia de la red.

Sistemas de Recomendación: Los sistemas de recomendación, como los utilizados por Netflix o Amazon, pueden utilizar variantes del algoritmo de Dijkstra para encontrar productos o contenidos relacionados.

¿Cómo lo implementarías el algoritmo Dijkstra en tu vida?

Aunque no siempre se aplica de manera directa en la vida cotidiana, sus conceptos pueden ser útiles en situaciones prácticas. Aquí hay algunas formas en las que podría considerar la implementación de los principios del algoritmo de Dijkstra en la vida diaria:

Optimización de Tiempo: En la gestión del tiempo y la programación de actividades, puedo aplicar principios de Dijkstra para priorizar tareas y actividades, asegurándose de que las actividades más importantes se realicen primero.

Optimización de Costos: En decisiones financieras, como la selección de inversiones o la administración del presupuesto, puedo utilizarse enfoques similares a Dijkstra para minimizar costos y maximizar rendimientos.

Redes Sociales y Relaciones Personales: En el mantenimiento de relaciones y redes sociales, puedo aplicar conceptos de Dijkstra para identificar relaciones valiosas y conexiones mutuas que puedan ser beneficiosas.

Ejercicio y Salud: Al planificar una rutina de ejercicios, puedo aplicar la lógica de Dijkstra para encontrar la ruta óptima hacia mis objetivos de salud y bienestar, optimizando el tiempo y los recursos.

Planificación de Carrera: Al tomar decisiones en mi carrera, puedo usar principios de Dijkstra para evaluar opciones, identificar caminos de desarrollo y priorizar las acciones que me acercarán a mis metas profesionales.

Ruta Diaria al Trabajo o Estudio: En la planificación de la ruta diaria al trabajo o estudio, puedo utilizar aplicaciones de mapas y navegación que se basan en algoritmos de rutas más cortas, que a menudo incluyen elementos de Dijkstra.

Optimización de Tareas Cotidianas: Al organizar mi día a día, puedo aplicar la idea de priorización basada en la importancia y urgencia de las tareas, similar a cómo Dijkstra evalúa las rutas más cortas.

¿Cómo lo implementarías en tu trabajo o tu trabajo de ensueño?

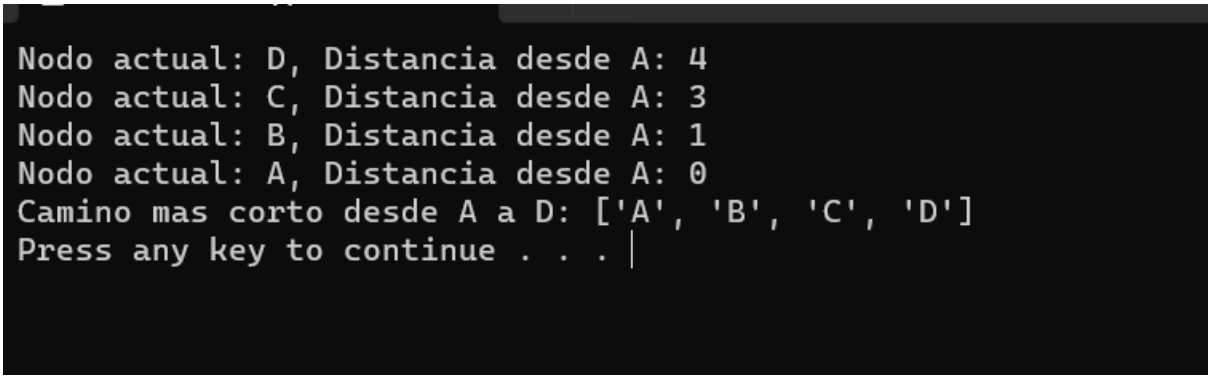
Después de esta investigación creo que lo importante aquí es que siempre seamos más productivos o tomarlo como ejemplo en nuestra vida por que uno como persona si necesita y más si nos ayuda a crecer como ingeniero, esto pues nos hará plantearnos cada problema que se nos presente y ver cómo resolverlo de la mejor manera, creo que en mi trabajo que es lo más importante de un ingeniero que es optimizando todo proceso para realizarlo lo más rápido y bien hecho posible, me ayudara a ver las posibles soluciones

para mi trabajo, como mi tiempo y ser más productivo lo cual si siempre hay mejora creceras como trabajador y escalaras a un mejor puesto.

Simulador en un programa

```python
import heapq
 # Inicializamos las distancias mínimas y el conjunto de nodos visitados.
def dijkstra(graph, start):
    distances = {node: float('inf') for node in graph}
    distances[start] = 0
    previous_nodes = {}
    priority_queue = [(0, start)]

# Encontramos el nodo no visitado con la distancia mínima.
    while priority_queue:
        current_distance, current_node = heapq.heappop(priority_queue)
        # Calculamos las distancias mínimas desde el nodo actual.
        if current_distance > distances[current_node]:
            continue

        for neighbor, weight in graph[current_node].items():
            distance = current_distance + weight
            # Actualizamos la distancia mínima si encontramos un camino más corto.
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                previous_nodes[neighbor] = current_node
                heapq.heappush(priority_queue, (distance, neighbor))

    return distances, previous_nodes

def shortest_path(graph, start, end):
    distances, previous_nodes = dijkstra(graph, start)
    path = []
```

```python
def shortest_path(graph, start, end):
    distances, previous_nodes = dijkstra(graph, start)
    path = []
    while end:
        path.insert(0, end)
        end = previous_nodes.get(end)
    return path

graph = {
    'A': {'B': 1, 'C': 4},
    'B': {'A': 1, 'C': 2, 'D': 5},
    'C': {'A': 4, 'B': 2, 'D': 1},
    'D': {'B': 5, 'C': 1}
}

start_node = 'A'
end_node = 'D'
distances, previous_nodes = dijkstra(graph, start_node)

current_node = end_node
while current_node:
    print(f'Nodo actual: {current_node}, Distancia desde {start_node}: {distances[current_node]}')
```



https://github.com/SaulCuenca/IA-P2/tree/main/PR3

References

Algoritmo de Dijkstra: ¿Qué es y para qué sirve? (n.d.). TFG Online. Retrieved October 26, 2023, from https://tfgonline.es/algoritmo-de-dijkstra/

Cassingena, E. (2022, October 24). *Algoritmo de la ruta más corta de Dijkstra - Introducción gráfica y detallada*. freeCodeCamp. Retrieved October 26, 2023, from https://www.freecodecamp.org/espanol/news/algoritmo-de-la-ruta-mas-corta-de-dijkstra-introduccion-grafica/