



INTELIGENCIA ARTIFICIAL

Antonio de Jesús Covarrubias Sánchez.

Registro: 22110347

T-6E1

Práctica 3

Algoritmo de Dijkstra.

03 DE NOVIEMBRE DEL 2024

CENTRO DE ENSEÑANZA TECNICA INDUSTRIAL

ALGORITMO DE DIJKSTRA

¿QUÉ ES?

El algoritmo de Dijkstra es un método desarrollado por el científico Edsger Dijkstra en 1956, diseñado para encontrar la ruta más corta desde un nodo de inicio a todos los demás nodos en un grafo ponderado. Los pesos en las aristas del grafo representan el costo o la distancia entre los nodos. Este algoritmo es un ejemplo clásico de algoritmos voraces, ya que en cada paso selecciona la opción más prometedora sin revisar todas las posibles soluciones.

¿PARA QUÉ SIRVE?

El algoritmo de Dijkstra se utiliza para resolver problemas de optimización relacionados con la búsqueda de rutas más cortas. Algunos de sus usos incluyen:

1. Navegación GPS:

Encontrar la ruta más eficiente entre dos ubicaciones.

2. Redes de telecomunicaciones:

Optimización de rutas para la transmisión de datos.

3. Planificación de rutas en logística:

Minimizar los costos de transporte o tiempo.

4. Juegos de video:

Determinar caminos óptimos para personajes o elementos dentro de un juego.

5. Sistemas de tráfico:

Control y gestión del flujo vehicular en tiempo real.

¿CÓMO SE IMPLEMENTA EN EL MUNDO?

- Sistemas de navegación:

Aplicaciones como Google Maps o Waze usan algoritmos similares a Dijkstra para calcular rutas óptimas.

- Redes informáticas:

Protocolos de enrutamiento como OSPF (Open Shortest Path First) utilizan versiones adaptadas del algoritmo para determinar las rutas más eficientes para la transmisión de paquetes de datos.

- Infraestructura de transporte:

Planificación y gestión de redes ferroviarias, aéreas y de tráfico terrestre.

- Robótica:

Navegación de robots autónomos en entornos complejos.

¿CÓMO LO IMPLEMENTARÍAS EN TU VIDA?

En la vida cotidiana, el concepto detrás del algoritmo de Dijkstra puede ser aplicado en:

1. Gestión de tiempo:

Optimizar rutas para realizar múltiples tareas, como elegir el camino más corto para hacer mandados.

2. Planificación de viajes:

Elegir itinerarios que minimicen el costo o tiempo de viaje entre múltiples destinos.

3. Toma de decisiones:

Evaluar opciones de inversión o planificación financiera considerando costos y beneficios para obtener el camino más rentable.

¿CÓMO LO IMPLEMENTARÍAS EN TU TRABAJO O TU TRABAJO DE ENSUEÑO? (PROGRAMACIÓN DE ROBOTS)

En la programación de robots, el algoritmo de Dijkstra se implementaría en:

1. Navegación autónoma:

- Los robots móviles, como drones o vehículos autónomos, podrían utilizar Dijkstra para navegar en entornos desconocidos o dinámicos, encontrando el camino más corto entre puntos de interés mientras evitan obstáculos.

2. Optimización de tareas en almacenes:

- Robots en almacenes automatizados pueden usar este algoritmo para minimizar el tiempo de recorrido al recoger y entregar paquetes.

3. Planificación de movimientos en robótica industrial:

- En robots manipuladores, como brazos robóticos, Dijkstra puede ayudar a calcular trayectorias eficientes para moverse entre diferentes posiciones de trabajo, evitando colisiones y minimizando el tiempo de operación.

4. Aplicaciones en robótica colaborativa:

- En entornos donde múltiples robots trabajan juntos, el algoritmo puede coordinar sus movimientos para evitar congestiones y optimizar la distribución de tareas.

CONCLUSIÓN

El algoritmo de Dijkstra es una herramienta poderosa para resolver problemas de optimización en diversos campos, desde navegación y logística hasta robótica avanzada. Implementar este algoritmo en proyectos de programación de robots no solo mejora la eficiencia, sino que también promueve soluciones inteligentes en la automatización y navegación autónoma.

BIBLIOGRAFÍAS

1. Cormen, TH, Leiserson, CE, Rivest, RL y Stein, C. (2009). *Introducción a los algoritmos* (3ª ed.). Prensa del MIT.
2. Dijkstra, EW (1959). Una nota sobre dos problemas relacionados con los gráficos. *Numerische Mathematik*, 1 (1), 269–271.
3. Kleinberg, J. y Tardos, E. (2006). *Diseño de algoritmos* . Pearson.
4. Sedgewick, R., y Wayne, K. (2011). *Algoritmos* (4.ª ed.). Addison-Wesley Professional.
5. Sniedovich, M. (2010). *Programación dinámica: fundamentos y principios* (2.ª ed.). CRC Press.