

ALGORITMO DE DIJKSTRA

El algoritmo de Dijkstra es un método para encontrar el camino más corto desde un nodo de origen a todos los demás nodos en un grafo ponderado, siempre y cuando los pesos de las aristas sean no negativos. Se basa en un proceso iterativo que asigna una distancia acumulada a cada nodo, actualizándolas gradualmente hasta que se encuentra la ruta mínima a todos los vértices.

¿Cómo funciona?

El algoritmo se desarrolla de la siguiente manera:

1. Inicialización:

Se establece la distancia del nodo de inicio a sí mismo como 0.

La distancia de todos los demás nodos se inicializa como "infinito".

Todos los nodos se marcan como "no visitados".

2. Iteración:

Se selecciona el nodo "no visitado" con la distancia acumulada más corta desde el inicio.

Este nodo se marca como "visitado" y se considera su distancia como definitiva.

Para todos los nodos adyacentes al nodo recién visitado, se evalúa si el camino a través de él es más corto que el camino actualmente conocido para ese nodo adyacente. Si es así, se actualiza la distancia del nodo adyacente y se registra el nodo desde el que se llega.

3. Finalización:

El proceso se repite hasta que todos los nodos son visitados o se alcanza el nodo de destino deseado.

Usos comunes

Navegación GPS: Para encontrar la ruta más corta entre dos puntos.

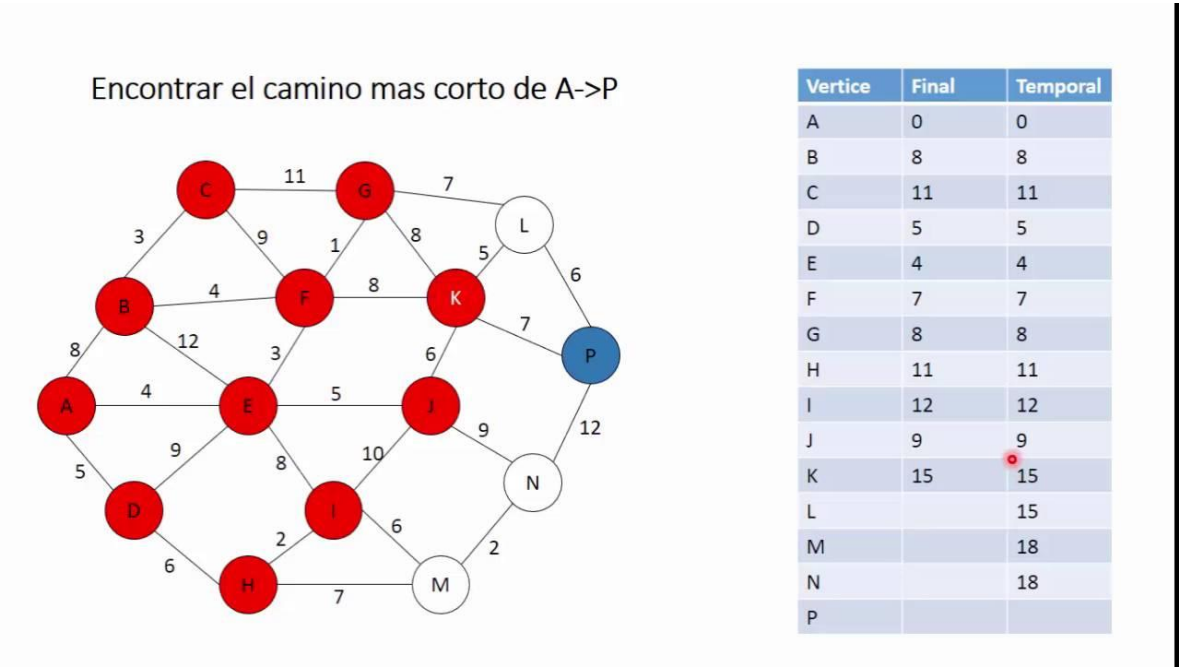
Redes de comunicación: Para determinar el camino más eficiente para el enrutamiento de datos (por ejemplo, en el protocolo OSPF).

Planificación de proyectos: Para identificar tareas críticas y el camino más largo (camino crítico) que determina la duración total de un proyecto.

Consideraciones

El algoritmo solo funciona para grafos con pesos de arista no negativos.

Un nuevo algoritmo publicado en 2025 ha logrado superar los límites de eficiencia impuestos por el algoritmo de Dijkstra en ciertos contextos, como los grafos dirigidos con pesos no negativos.



Ejecuciones

```
PS D:\jrubi\Documents\TECNOLOGICO\9no Semestre\Visual Studio> & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\jrubi\AppData\Roaming\Code\User\workspaceStorage\62aa3ecdf8d0f347347f46e9b5b94086\redhat.java\jdt_ws\Visual Studio_aadd6e21\bin' 'MainGrafo'
Nodo 1 conectado a: (2, peso: 5) (3, peso: 2)
Nodo 2 conectado a: (1, peso: 5) (3, peso: 1) (4, peso: 5)
Nodo 3 conectado a: (1, peso: 2) (2, peso: 1) (6, peso: 1)
Nodo 4 conectado a: (2, peso: 5) (6, peso: 5) (5, peso: 5)
Nodo 5 conectado a: (6, peso: 2) (4, peso: 5)
Nodo 6 conectado a: (3, peso: 1) (4, peso: 5) (5, peso: 2)
¿Desde qué nodo quieres iniciar el recorrido?
3
¿A qué nodo quieres llegar desde el nodo 3? 5
Distancia mínima desde el nodo 3 al nodo 5: 3
Camino: 3 -> 6 -> 5
PS D:\jrubi\Documents\TECNOLOGICO\9no Semestre\Visual Studio>
```

```
Storage\62aa3ecdf8d0f347347f46e9b5b94086\redhat.java\jdt_ws\Visual Studio_aadd6e21\bin' 'MainGrafo'
Nodo 1 conectado a: (2, peso: 5) (3, peso: 2)
Nodo 2 conectado a: (1, peso: 5) (3, peso: 1) (4, peso: 5)
Nodo 3 conectado a: (1, peso: 2) (2, peso: 1) (6, peso: 1)
Nodo 4 conectado a: (2, peso: 5) (6, peso: 5) (5, peso: 5)
Nodo 5 conectado a: (6, peso: 2) (4, peso: 5)
Nodo 6 conectado a: (3, peso: 1) (4, peso: 5) (5, peso: 2)
¿Desde qué nodo quieres iniciar el recorrido?
1
¿A qué nodo quieres llegar desde el nodo 1? 4
Distancia mínima desde el nodo 1 al nodo 4: 8
Camino: 1 -> 3 -> 2 -> 4
PS D:\jrubi\Documents\TECNOLOGICO\9no Semestre\Visual Studio>
```