La cantidad de aristas salientes se llama grado positivo del vértice.

La cantidad de aristas entrantes se llama grado negativo del vértice.

Un vértice que no tiene aristas salientes ni entrantes es un vértice aislado.

Se dice que un vértice tiene aristas entrantes, cuando existe al menos una arista que tiene al vértice en cuestión como destino. Se dice que un vértice tiene aristas salientes, cuando existe al menos una arista que tiene al vértice en cuestión como origen.

En otras palabras, un vértice v tiene aristas entrantes si existe al menos una arista **(x, y) ∈ A con v = y**; tiene aristas salientes si existe al menos una arista **(x, y) ∈ A con v = x**

El vértice B es adyacente al vértice A si existe una arista (A, B) ∈ A que los comunica. En el caso de grafos dirigidos, puede suceder que B sea adyacente de A, pero no recíprocamente.

Un vértice es alcanzable desde otro si existe un camino que una al segundo con el primero.

Un ciclo es un camino que comienza y termina en el mismo vértice.

Algoritmo Dijkstra:

El algoritmo de camino mínimo de único origen, también conocido como algoritmo de Dijkstra, es un algoritmo de grafos que encuentra la ruta más corta entre un nodo de origen y todos los demás nodos en un grafo ponderado, siempre que los pesos de las aristas sean no negativos.

Busca el camino más corto a todos los vértices de un grafo.

La cantidad de vértices la podemos escribir como: n = |V|

Donde n es la cantidad de vértices

Costo: N x N = N2

Donde N es la cantidad de vértices y N la cantidad de vértices relajados

O(N2)

O(|V|2)

Parte Complejidad IMPLEMENTACION JAVA

Inicialización O(n)

Selección de mínimo O(n²)

Revisión de vecinos O(n²)

Total O(n²)

El algoritmo de Dijkstra tiene una complejidad temporal de O(n²) cuando se implementa utilizando una matriz de adyacencia y búsqueda lineal del vértice de menor distancia, como es habitual en implementaciones estáticas. Esto se debe a que en cada iteración del algoritmo se selecciona el vértice no visitado con la menor distancia actual, lo cual requiere recorrer todos los vértices (O(n)), y luego se relajan las aristas hacia sus vecinos, lo que también implica recorrer potencialmente todos los vértices (O(n)). Como estas operaciones se repiten hasta n veces (una por vértice), el costo total resulta en O(n × n) = O(n²). Esta complejidad es adecuada para grafos pequeños o representaciones simples, pero puede volverse ineficiente en grafos grandes o dispersos.

¿Como funciona?

Inicialización:

* Se asigna una distancia inicial infinita a todos los nodos, excepto al nodo de origen, que se inicializa con distancia cero.
* Se crea un conjunto de nodos no visitados, que inicialmente incluye todos los nodos.

Iteración:

* Se selecciona el nodo con la menor distancia actual entre los nodos no visitados.
* Se marca el nodo seleccionado como visitado.
* Se actualizan las distancias de los nodos adyacentes al nodo seleccionado: si la distancia actual desde el nodo de origen a un nodo adyacente, pasando por el nodo seleccionado, es menor que la distancia actual de ese nodo, se actualiza la distancia.

Repetición:

* Se repite el paso 2 hasta que todos los nodos hayan sido visitados.