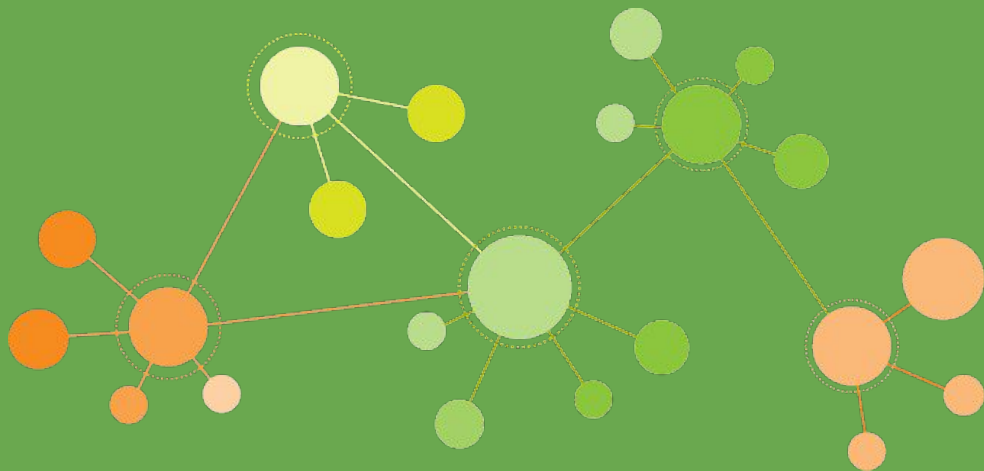

GRAFOS

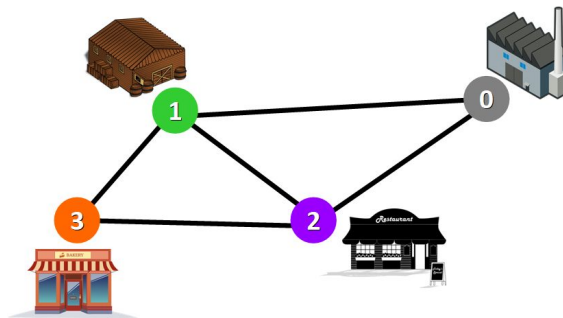
Yaneth Mejía Rendón



Grafos

Los grafos se pueden aplicar a escenarios de la vida real.

Por ejemplo, podríamos modelar una red de transporte, en la cual los nodos representan instalaciones para enviar o recibir productos y los arcos representan caminos que los conectan.

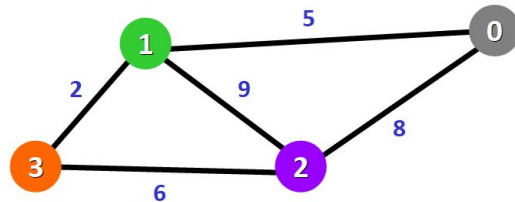
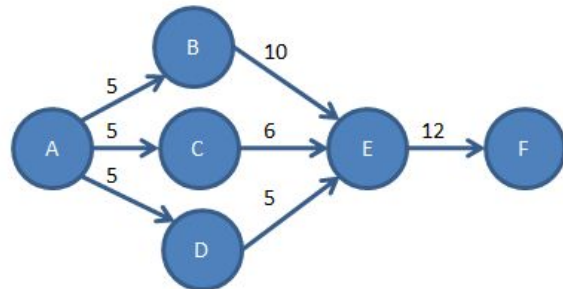


Grafos ponderados

Es un grafo cuyos arcos tienen un "peso", "valor", o "costo" asociado.

El valor de cada arco puede representar la distancia, tiempo, u otro valor que modele la conexión entre el par de nodos que conecta.

Pueden ser ponderados dirigidos, ponderados no dirigidos y sus valores pueden variar entre positivos y negativos.

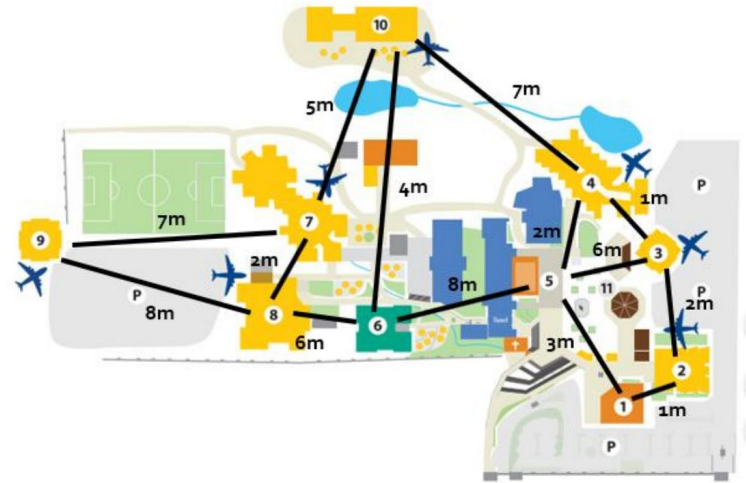


Algoritmo Dijkstra

Con el algoritmo se puede encontrar la ruta más corta o el camino más corto entre los nodos de un grafo.

Siempre se parte desde un nodo origen a todos los otros nodos del grafo, generando un árbol del camino más corto.

Este algoritmo es usado por los dispositivos GPS para encontrar el camino más corto entre la ubicación actual y el destino del usuario.



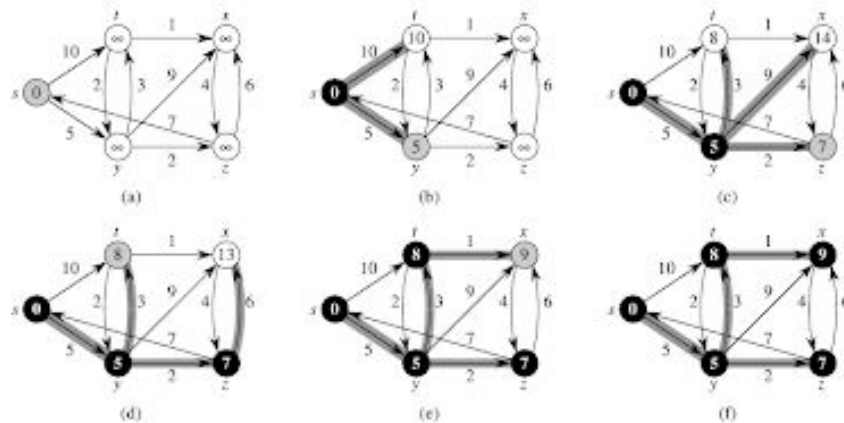
Algoritmo Dijkstra

- El algoritmo inicia en el nodo que escojas (el nodo de origen) y analiza el grafo para encontrar el camino más corto entre ese nodo y todos los otros nodos en el grafo.
- El algoritmo mantiene un registro de la distancia conocida más corta desde el nodo de origen hasta cada nodo y actualiza el valor si encuentra un camino más corto.
- Una vez se ha encontrado el camino más corto entre el nodo de origen y otro nodo, ese nodo se marca como "visitado" y se agrega al camino.
- El proceso continúa hasta que todos los nodos en el grafo han sido añadidos al camino.

Algoritmo Dijkstra

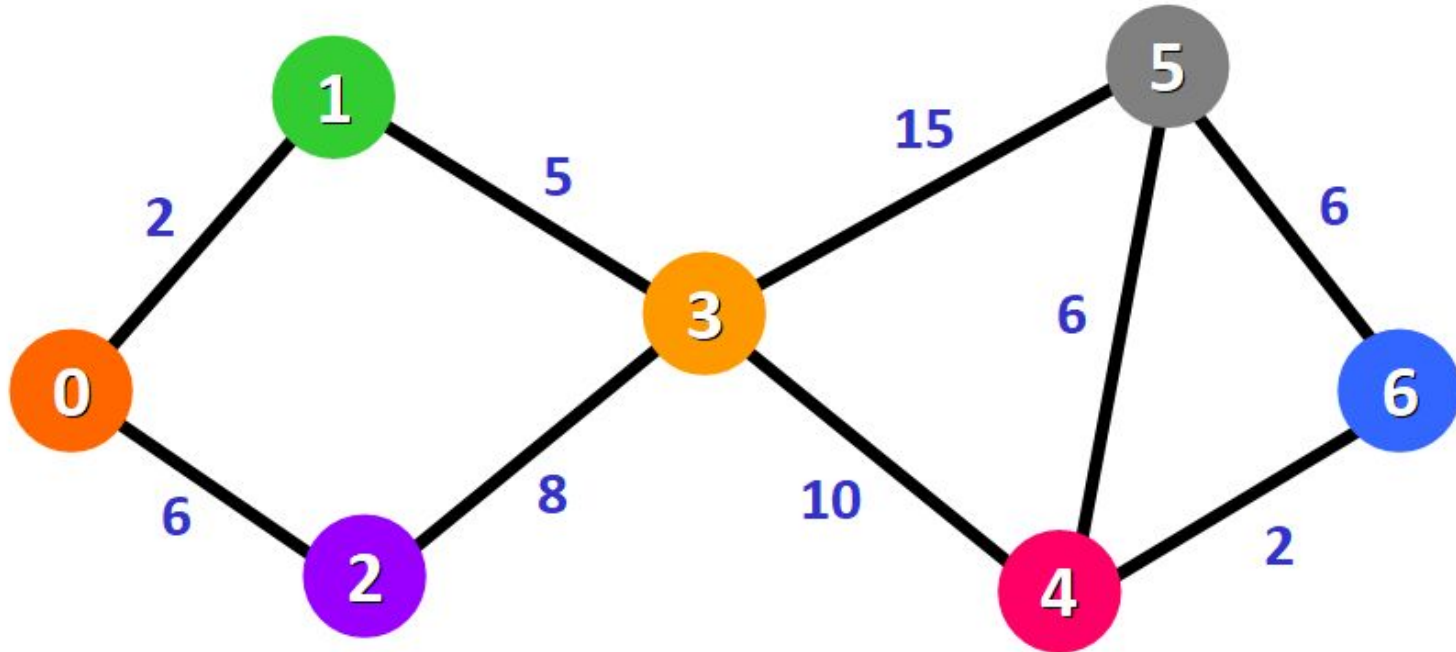
Solo puede aplicarse a grafos con arcos cuyos valores o pesos son positivos. Debido a que los valores de los arcos deben ser sumados para encontrar el camino más corto.

Si existe un valor negativo en el grafo, el algoritmo no funcionará correctamente.



Ejemplo

Se tiene el siguiente grafo



Ejemplo

El algoritmo generará el camino más corto desde el nodo 0 hasta todos los demás nodos del grafo.

Tendremos esta lista de distancias:

- La distancia desde el nodo de origen a sí mismo es 0. En este ejemplo, el nodo de origen será el nodo 0, pero puede ser cualquier nodo que escojas.
- También tendremos esta lista para mantener un registro de los nodos que todavía no se han visitado.

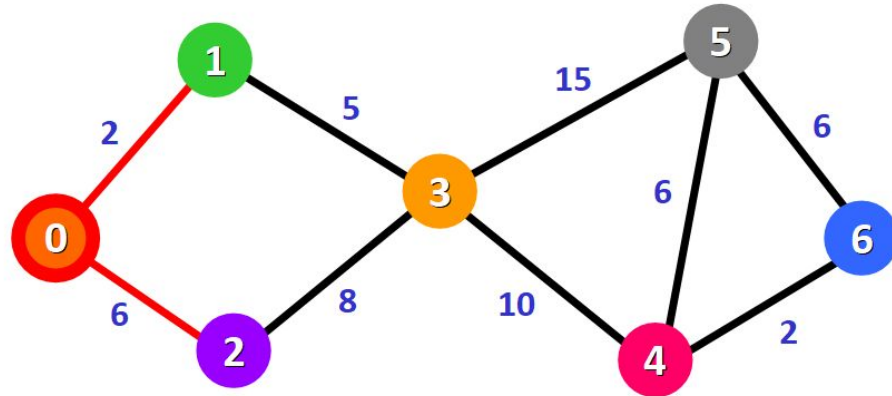
Nodos por visitar: {0, 1, 2, 3, 4, 5, 6}

Ejemplo

Cómo escogemos iniciar en el nodo 0, podemos marcar este nodo como visitado.

~~0~~, 1, 2, 3, 4, 5, 6

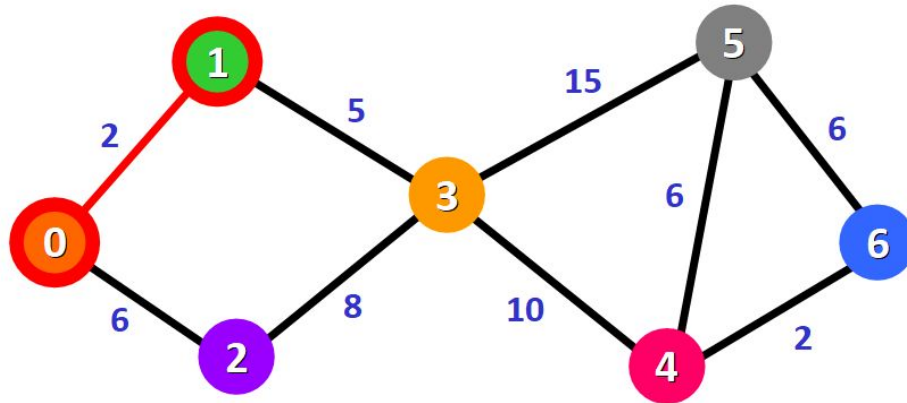
Ahora debemos verificar la distancia desde el nodo 0 hasta sus nodos adyacentes. Estos nodos son el nodo 1 y el nodo 2:



Ejemplo

Si verificamos la lista de distancias, podemos ver que el nodo 1 tiene la menor distancia hasta el nodo de origen, así que lo añadimos al camino.

En el diagrama, podemos representar esto con un borde rojo:

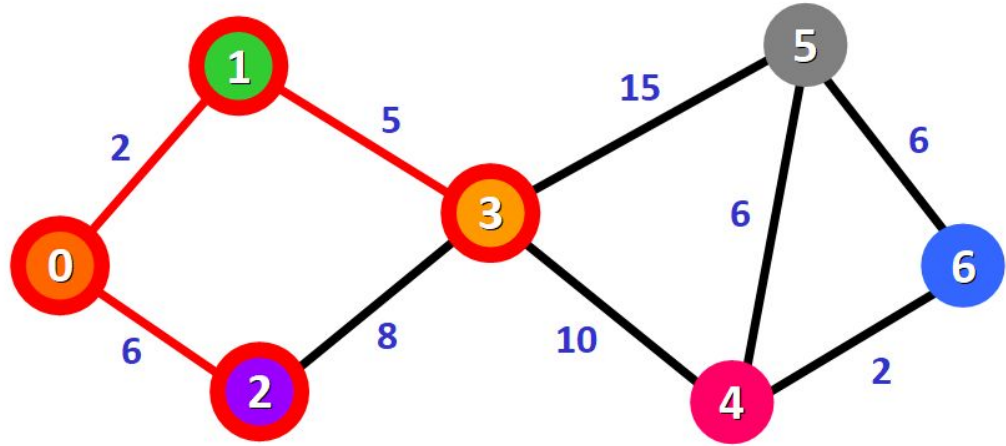


Ejemplo

Los nodos 3 y 2 son adyacentes a los nodos que ya están en el camino porque están directamente conectados al nodo 1 y al nodo 0, respectivamente.

Se deben sumar sus valores:

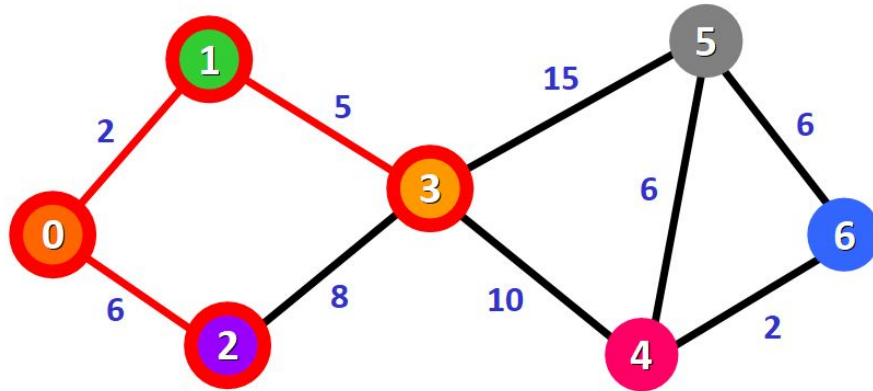
- $2 + 5 = 7$
- $6 + 8 = 14$



Ejemplo

Ahora repitamos el proceso nuevamente.

Debemos verificar los nuevos nodos adyacentes que no hemos visitado hasta el momento. Estos nodos son el nodo 4 y el nodo 5 ya que son adyacentes al nodo 3.

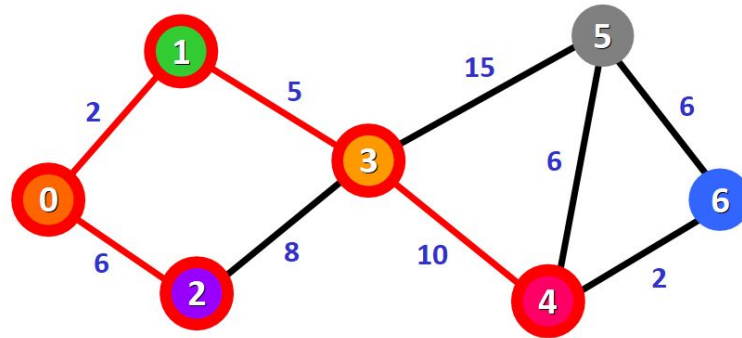


Ejemplo

Se repite el proceso y se elige el nodo 4

Existen tres posibles caminos que podemos escoger para alcanzar el nodo 5:

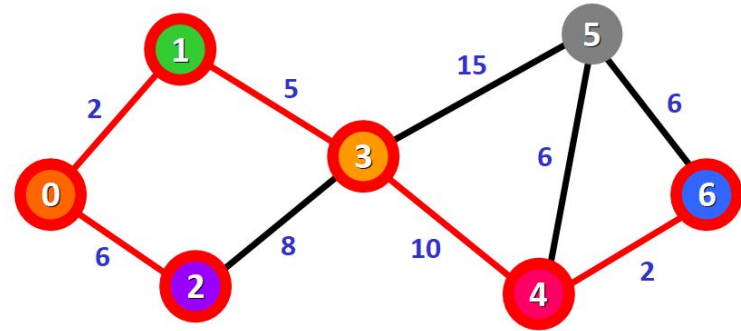
- **Opción 1: 0 -> 1 -> 3 -> 5 con una distancia de 22 ($2 + 5 + 15$).**
- Opción 2: 0 -> 1 -> 3 -> 4 -> 5 con una distancia de 23 ($2 + 5 + 10 + 6$).
- Opción 3: 0 -> 1 -> 3 -> 4 -> 6 -> 5 con una distancia de 25 ($2 + 5 + 10 + 2 + 6$).



Ejemplo

Se repite el proceso y se elige el nodo 6. Las líneas rojas indican los arcos que pertenecen al camino más corto.

Debes seguir estos arcos para seguir el camino más corto para alcanzar un nodo en el grafo, si inicias el recorrido desde el nodo 0.



0 -> 1 -> 3 -> 4 -> 6