

## **Instituto Politécnico Nacional**

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE  
INGENIERÍA CAMPUS ZACATECAS

### **REPORTE PRÁCTICA 03: IMPLEMENTACIÓN Y EVALUACIÓN DEL ALGORITMO DE DIJKSTRA**

*Docente: M. en C. Erika Sánchez-Femat*

Lucia Iveth De La Vega Hernández  
3CM2

01 Diciembre 2023

## Introducción

La resolución del problema de encontrar los caminos mínimos en un grafo ponderado es esencial en diversos contextos, desde redes de transporte hasta sistemas de información. El algoritmo de Dijkstra es una herramienta valiosa en este sentido, y en esta práctica, se explorará su funcionamiento y lo se implementará en Python.

## Desarrollo de la Práctica

### Implementación del Algoritmo de Dijkstra

Define una función `dijkstra` que tome el grafo y un vértice de inicio como entrada y devuelva los caminos mínimos desde el vértice de inicio a todos los demás vértices. Puedes utilizar un diccionario para mantener un registro de las distancias mínimas. Nuestro algoritmo intenta encontrar la ruta más corta de un vértice inicial a uno final.

Se define el grafo en el cual se agregan los vertices y las aristas con su peso respectivo para después llamar a la función `Dijkstra` en el cual encontrará todos los posibles caminos con su respectiva distancia para al final elegir el camino más corto entre los vértices.

### Generación de Casos de Prueba de los 5 Grafos

Para comprender mejor el funcionamiento de este algoritmo, se ejecutó con 5 grafos distintos y se tomó el tiempo de ejecución respectivo de cada algoritmo.

En la siguiente página se muestran los 5 grafos utilizados para los casos prueba.

### Medición del Tiempo

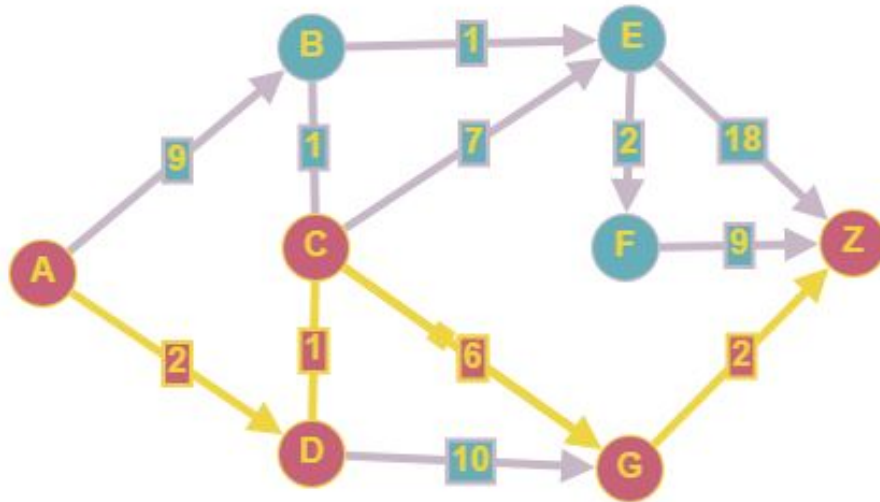
Deberás escribir una función que mida el tiempo de ejecución para cada caso de prueba.

Esto se realizó mediante la librería `time` y en cada grafo se fue midiendo el tiempo que tardó en ejecutarse.

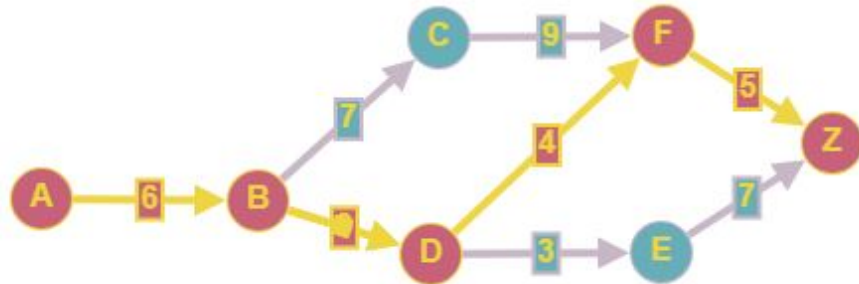
### Visualización de Datos

También se muestra el resultado de los 5 grafos donde se obtiene su camino más corto y su distancia, en conjunto con el tiempo de ejecución.

La longitud del camino más corto es 11:  $A \Rightarrow D \Rightarrow C \Rightarrow G \Rightarrow Z$

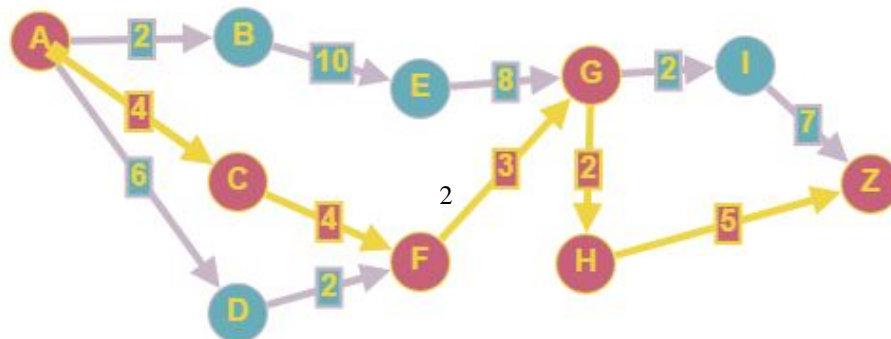


La longitud del camino más corto es 24:  $A \Rightarrow B \Rightarrow D \Rightarrow F \Rightarrow Z$

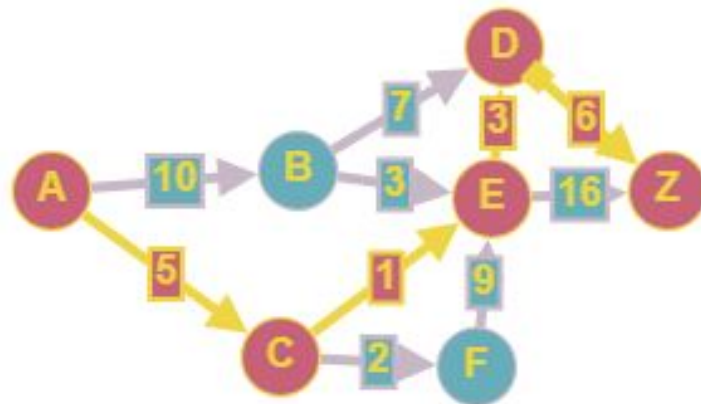


La longitud del camino más corto es 18:  $A \Rightarrow C \Rightarrow F \Rightarrow G \Rightarrow H \Rightarrow Z$

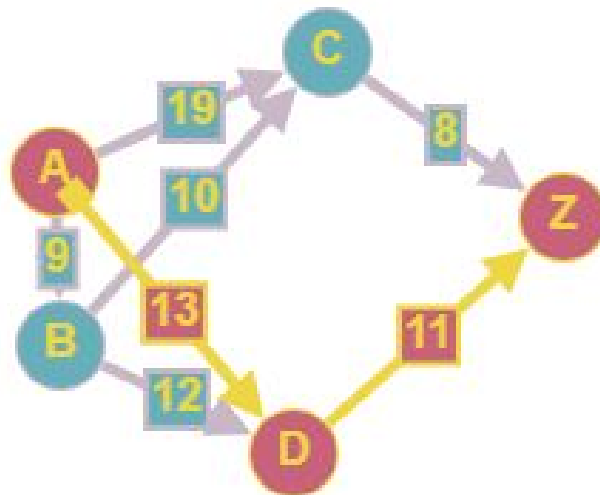
Rep



La longitud del camino más corto es 15:  $A \Rightarrow C \Rightarrow E \Rightarrow D \Rightarrow Z$



La longitud del camino más corto es 24:  $A \Rightarrow D \Rightarrow Z$



```

-----GRAFO 1-----
***Camino mínimo:
Camino1=['a', 'd', 'c', 'b', 'e', 'f', 'g', 'z']
***Distancia del Grafo:
Distancia1={'a': 0, 'b': 4, 'c': 3, 'd': 2, 'e': 5, 'f': 7, 'g': 9, 'z': 11}
Tiempo 1: 9.226799011230469e-05
-----GRAFO 2-----
***Camino mínimo:
Camino2=['a', 'b', 'c', 'd', 'e', 'f', 'z']
***Distancia del Grafo:
Distancia2={'a': 0, 'b': 6, 'c': 13, 'd': 15, 'e': 18, 'f': 19, 'z': 24}
Tiempo 2: 4.267692565917969e-05
-----GRAFO 3-----
***Camino mínimo:
Camino3=['a', 'b', 'c', 'd', 'f', 'g', 'e', 'h', 'i', 'z']
***Distancia del Grafo:
Distancia3={'a': 0, 'b': 2, 'c': 4, 'd': 6, 'e': 12, 'f': 8, 'g': 11, 'h': 13, 'i': 13, 'z': 18}
Tiempo 3: 5.316734313964844e-05
-----GRAFO 4-----
***Camino mínimo:
Camino4=['a', 'c', 'e', 'f', 'd', 'b', 'z']
***Distancia del Grafo:
Distancia4={'a': 0, 'b': 10, 'c': 5, 'd': 9, 'e': 6, 'f': 7, 'z': 15}
Tiempo 4: 4.00543212890625e-05
-----GRAFO 5-----
***Camino mínimo:
Camino5=['a', 'b', 'd', 'c', 'z']
***Distancia del Grafo:
Distancia5={'a': 0, 'b': 9, 'c': 19, 'd': 13, 'z': 24}
Tiempo 5: 5.650520324707031e-05

```

## Entrega de Resultados

Deberás desarrollar un reporte en Overleaf con Portada, Introducción, Desarrollo, Resultados y Conclusiones y Referencias. En la sección de desarrollo tendrás que explicar el procedimiento de implementación del algoritmo, con código e imágenes para apoyar la explicación. En la sección de Resultados se deberán agregar 5 casos de ejecución del algoritmo con imágenes y explicación de cada uno de los casos.

El código en python y el archivo PDF del reporte se deberán subir al repositorio que cada uno de los alumnos tiene en Github.

## **Conclusión**

En conclusión el algoritmo de Dijkstra es un método muy útil para encontrar caminos cortos de un punto a otro. Al implementarlo en Python es muy eficiente ya que directamente indica la distancia y el camino mínimo encontrado de manera rápida solamente indicando el grafo donde tiene sus vértices, conexiones(aristas) con su respectivo peso, dirección de las aristas, vertice inicial y final.