



Francisco Javier Enriquez Plascencia 21110354

Centro de Enseñanza Técnica Industrial

Inteligencia Artificial

6E

Maestro: Mauricio Cabrera

Practica 3,

¿Qué es el algoritmo de Dijkstra?

El algoritmo de Dijkstra es un algoritmo para encontrar el camino más corto entre un nodo dado (el nodo de origen) y todos los otros nodos del grafo. Este algoritmo usa los valores de los arcos para encontrar el camino que minimiza el valor total entre el nodo de origen y los demás nodos del grafo.

¿Para qué sirve el algoritmo de Dijkstra?

El algoritmo de Dijkstra se utiliza en una amplia gama de aplicaciones, que incluyen:

- Navegación: El algoritmo de Dijkstra se utiliza en sistemas de navegación para encontrar la ruta más corta entre dos puntos.
- Distribución de tráfico: El algoritmo de Dijkstra se utiliza para optimizar el tráfico en redes de carreteras o ferrocarriles.
- Estructuras de datos: El algoritmo de Dijkstra se utiliza para construir árboles de búsqueda binarios.

¿Cómo se implementa el algoritmo de Dijkstra en el mundo?

El algoritmo de Dijkstra se implementa en una amplia gama de productos y servicios, que incluyen:

- Navegadores GPS: Los navegadores GPS utilizan el algoritmo de Dijkstra para encontrar la ruta más corta entre dos puntos.
- Sistemas de transporte público: Los sistemas de transporte público utilizan el algoritmo de Dijkstra para optimizar las rutas de los autobuses y trenes.
- Software de diseño de redes: El software de diseño de redes utiliza el algoritmo de Dijkstra para encontrar el camino más corto entre dos puntos en una red.

¿Cómo implementaría el algoritmo de Dijkstra en mi vida?

Podría implementar el algoritmo de Dijkstra en mi vida para encontrar la ruta más corta entre dos puntos en mi ciudad. Por ejemplo, podría usar el algoritmo para encontrar la ruta más rápida para ir del trabajo a casa o para encontrar la ruta más corta para visitar a un amigo.

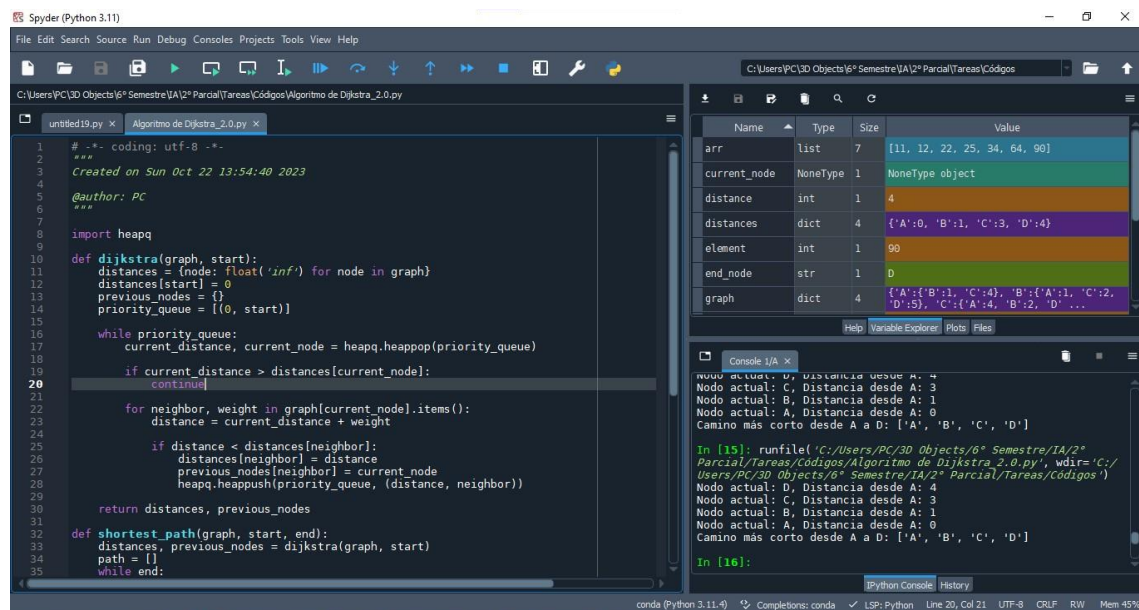
¿Cómo implementaría el algoritmo de Dijkstra en mi trabajo o en mi trabajo de ensueño?

En mi trabajo actual, podría utilizar el algoritmo de Dijkstra para optimizar las rutas de entrega de los productos. Por ejemplo, podría usar el algoritmo para encontrar la ruta más eficiente para entregar un paquete a un cliente.

En mi trabajo de ensueño, podría utilizar el algoritmo de Dijkstra para ayudar a las personas a encontrar el camino más corto para acceder a servicios esenciales, como atención médica o educación. Por ejemplo, podría usar el algoritmo para encontrar la ruta más rápida para llegar a un hospital o para encontrar la ruta más corta para llegar a una escuela.

En general, el algoritmo de Dijkstra es una herramienta poderosa que se puede utilizar para encontrar el camino más corto entre dos puntos en una amplia gama de aplicaciones.

Código



The screenshot shows the Spyder Python IDE with a file named 'Algoritmo de Dijkstra_2.0.py' open. The code implements Dijkstra's algorithm and a shortest path function. The console output shows the execution of the algorithm for a specific graph, displaying the current node, distances, and the shortest path found.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Oct 22 13:54:40 2023
4
5 @author: PC
6 """
7
8 import heapq
9
10 def dijkstra(graph, start):
11     distances = {node: float('inf') for node in graph}
12     distances[start] = 0
13     previous_nodes = {}
14     priority_queue = [(0, start)]
15
16     while priority_queue:
17         current_distance, current_node = heapq.heappop(priority_queue)
18
19         if current_distance > distances[current_node]:
20             continue
21
22         for neighbor, weight in graph[current_node].items():
23             distance = current_distance + weight
24
25             if distance < distances[neighbor]:
26                 distances[neighbor] = distance
27                 previous_nodes[neighbor] = current_node
28                 heapq.heappush(priority_queue, (distance, neighbor))
29
30     return distances, previous_nodes
31
32 def shortest_path(graph, start, end):
33     distances, previous_nodes = dijkstra(graph, start)
34     path = []
35     while end:
```

Variable Explorer:

Name	Type	Size	Value
arr	list	7	[11, 12, 22, 25, 34, 64, 90]
current_node	NoneType	1	NoneType object
distance	int	1	4
distances	dict	4	{'A':0, 'B':1, 'C':3, 'D':4}
element	int	1	90
end_node	str	1	D
graph	dict	4	{'A':{'B':1, 'C':4}, 'B':{'A':1, 'C':2, 'D':5}, 'C':{'A':4, 'B':2, 'D':...}}

Console I/A:

```
In [15]: runfile('C:/Users/PC/3D Objects/6º Semestre/IA/2º Parcial/Tareas/Códigos/Algoritmo de Dijkstra_2.0.py', wdir='C:/Users/PC/3D Objects/6º Semestre/IA/2º Parcial/Tareas/Códigos')
Nodo actual: D, Distancia desde A: 4
Nodo actual: C, Distancia desde A: 3
Nodo actual: B, Distancia desde A: 1
Nodo actual: A, Distancia desde A: 0
Camino más corto desde A a D: ['A', 'B', 'C', 'D']

In [16]:
```

Bibliografía

Navone, E. C. (2023). Algoritmo de la ruta más corta de Dijkstra - Introducción gráfica y detallada. *freeCodeCamp.org*. <https://www.freecodecamp.org/espanol/news/algoritmode-la-ruta-mas-corta-de-dijkstra-introducciongrafica/#:~:text=El%20algoritmo%20de%20Dijkstra%20encuentra,los%20demos%20los%20nodos%20del%20grafo.>