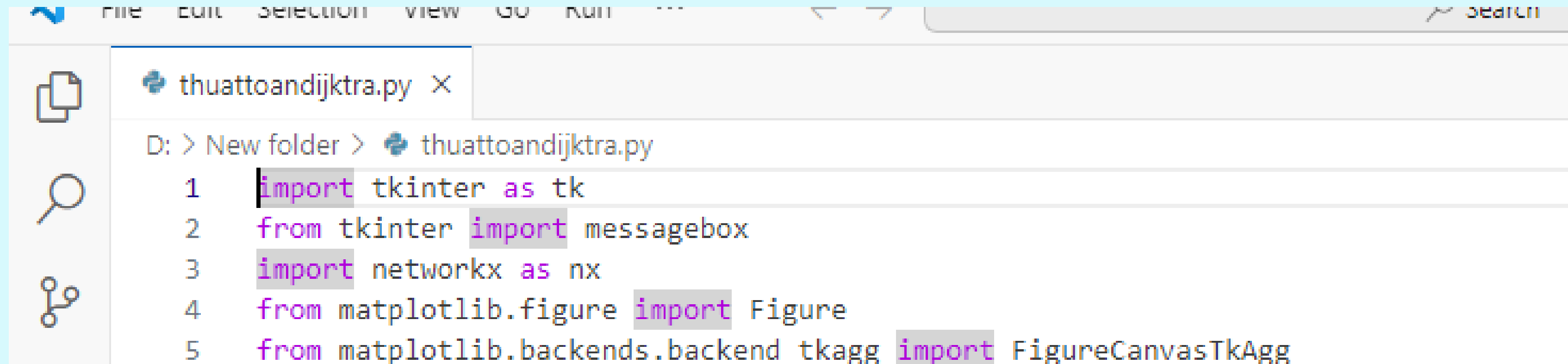




VIẾT CHƯƠNG TRÌNH MÔ PHỎNG THUẬT TOÁN DIJKTRA

THƯ VIỆN



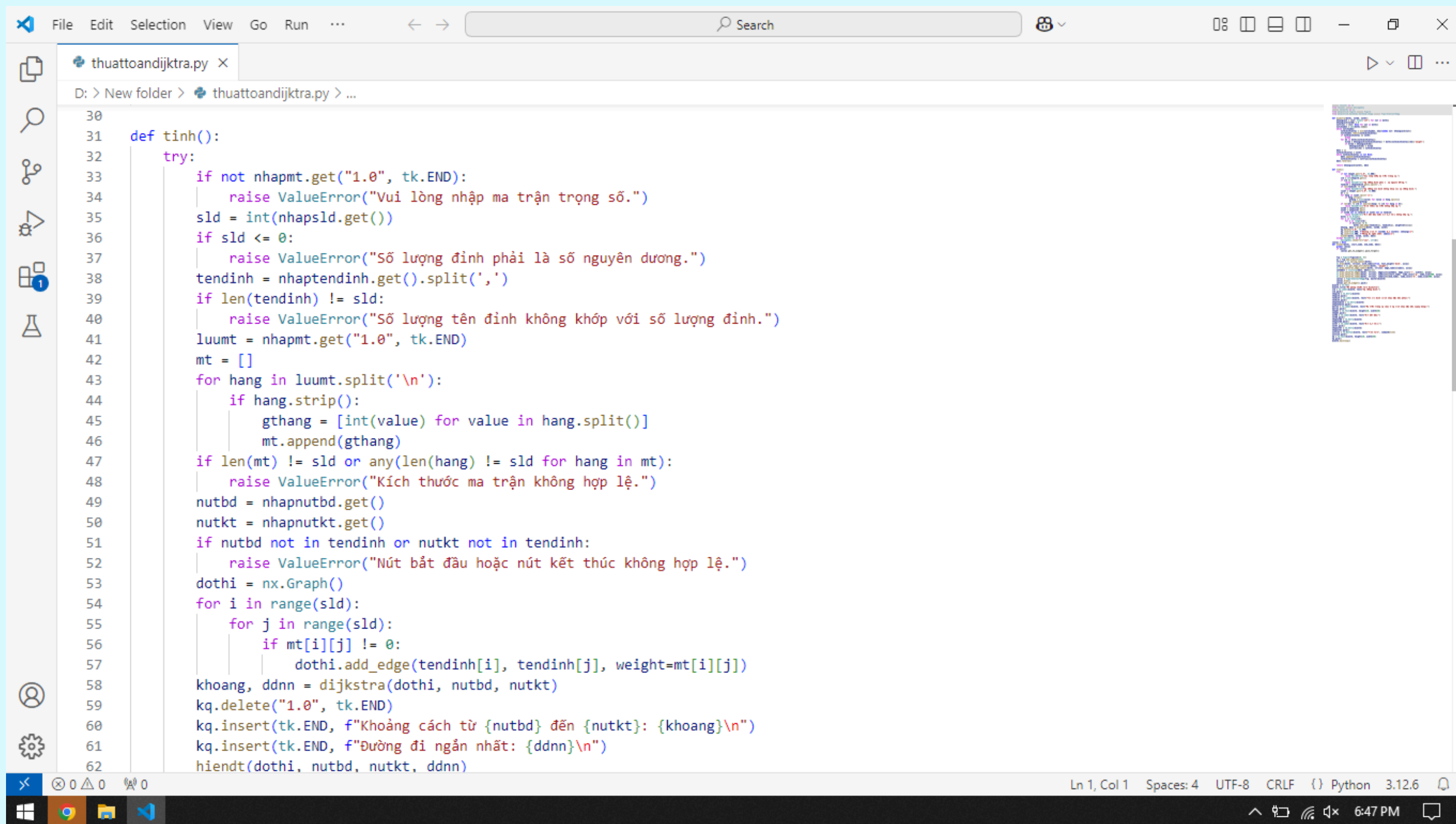
The image shows a screenshot of a code editor window. The title bar at the top includes a menu bar with options: file, edit, selection, view, go, run, and a search bar on the right. Below the title bar, there is a tab labeled 'thuattoandijkstra.py' with a close button. The main editing area shows the following Python code:

```
D: > New folder > thuattoandijkstra.py
1  import tkinter as tk
2  from tkinter import messagebox
3  import networkx as nx
4  from matplotlib.figure import Figure
5  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
```

HÀM DIJKTRA

```
6
7 def dijkstra(dothi, nutbd, nutkt):
8     khoangcach = {nut: float('inf') for nut in dothi}
9     khoangcach[nutbd] = 0
10    nuttruoc = {nut: None for nut in dothi}
11    nutchuaden = set(dothi.nodes)
12    while nutchuaden:
13        nutkcminhientai = min(nutchuaden, key=lambda nut: khoangcach[nut])
14        nutchuaden.remove(nutkcminhientai)
15        if nutkcminhientai == nutkt:
16            break
17        for ke in dothi[nutkcminhientai]:
18            kctam = khoangcach[nutkcminhientai] + dothi[nutkcminhientai][ke]['weight']
19            if kctam < khoangcach[ke]:
20                khoangcach[ke] = kctam
21                nuttruoc[ke] = nutkcminhientai
22    ddnn = []
23    nutkcminhientai = nutkt
24    while nutkcminhientai is not None:
25        ddnn.append(nutkcminhientai)
26        nutkcminhientai = nuttruoc[nutkcminhientai]
27    ddnn.reverse()
28
29    return khoangcach[nutkt], ddnn
30
```

HÀM TÍNH TOÁN



The screenshot shows a Python IDE with a file named `thuattoandijkstra.py`. The code implements a Dijkstra's algorithm for finding the shortest path between two nodes in a graph. The graph is represented as an adjacency list, where each node is a string representing a location, and the edges are weighted by distance. The algorithm uses a priority queue (implemented as a list) to select the next node to visit. The code is written in Python 3.12.6 and uses the `tkinter` module for user input and output.

```
30
31 def tinh():
32     try:
33         if not nhapmt.get("1.0", tk.END):
34             raise ValueError("Vui lòng nhập ma trận trọng số.")
35         sld = int(nhapsld.get())
36         if sld <= 0:
37             raise ValueError("Số lượng đỉnh phải là số nguyên dương.")
38         tendinh = nhaptendinh.get().split(',')
39         if len(tendinh) != sld:
40             raise ValueError("Số lượng tên đỉnh không khớp với số lượng đỉnh.")
41         luumt = nhapmt.get("1.0", tk.END)
42         mt = []
43         for hang in luumt.split('\n'):
44             if hang.strip():
45                 gthang = [int(value) for value in hang.split()]
46                 mt.append(gthang)
47         if len(mt) != sld or any(len(hang) != sld for hang in mt):
48             raise ValueError("Kích thước ma trận không hợp lệ.")
49         nutbd = nhapnutbd.get()
50         nutkt = nhapnutkt.get()
51         if nutbd not in tendinh or nutkt not in tendinh:
52             raise ValueError("Nút bắt đầu hoặc nút kết thúc không hợp lệ.")
53         dothi = nx.Graph()
54         for i in range(sld):
55             for j in range(sld):
56                 if mt[i][j] != 0:
57                     dothi.add_edge(tendinh[i], tendinh[j], weight=mt[i][j])
58         khoang, ddnn = dijkstra(dothi, nutbd, nutkt)
59         kq.delete("1.0", tk.END)
60         kq.insert(tk.END, f"Khoảng cách từ {nutbd} đến {nutkt}: {khoang}\n")
61         kq.insert(tk.END, f"Đường đi ngắn nhất: {ddnn}\n")
62         hiendt(dothi, nutbd, nutkt, ddnn)
```

File Edit Selection View Go Run ... Search

thuattoandijkstra.py X

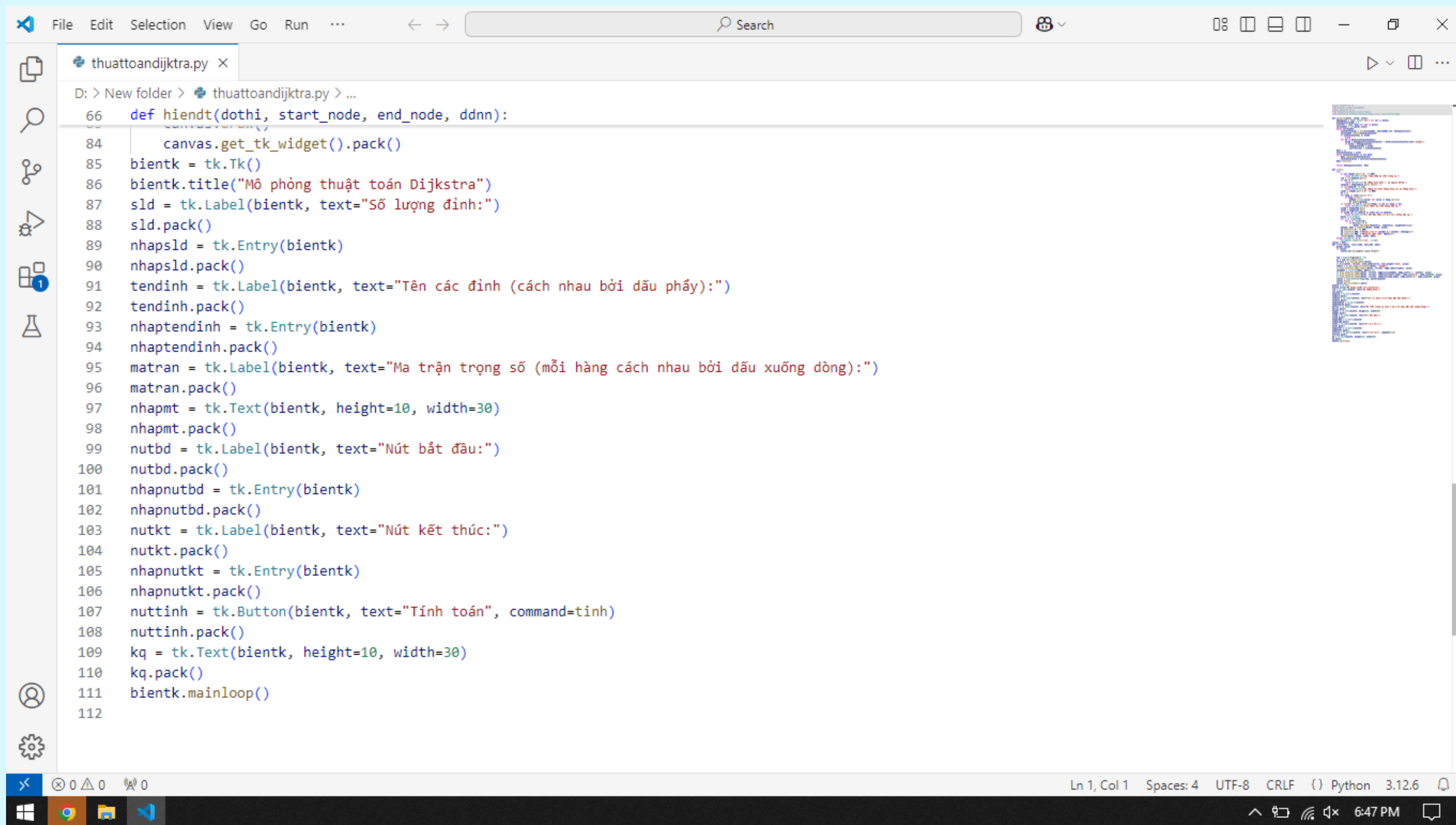
D: > New folder > thuattoandijkstra.py > ...

```
31 def tinh():
55     for j in range(sld):
56         if mt[i][j] != 0:
57             dothi.add_edge(tendinh[i], tendinh[j], weight=mt[i][j])
58     khoang, ddnn = dijkstra(dothi, nutbd, nutkt)
59     kq.delete("1.0", tk.END)
60     kq.insert(tk.END, f"Khoảng cách từ {nutbd} đến {nutkt}: {khoang}\n")
61     kq.insert(tk.END, f"Đường đi ngắn nhất: {ddnn}\n")
62     hiendt(dothi, nutbd, nutkt, ddnn)
63 except ValueError as e:
64     messagebox.showerror("Lỗi", str(e))
65 canvas = None
66 def hiendt(dothi, start_node, end_node, ddnn):
67     global canvas
68     if canvas:
69         canvas.get_tk_widget().pack_forget()
70
71
72     fig = Figure(figsize=(6, 4))
73     ax = fig.add_subplot(111)
74     vtrirut = nx.spring_layout(dothi)
75     nx.draw(dothi, vtrirut, with_labels=True, font_weight='bold', ax=ax)
76     labels = nx.get_edge_attributes(dothi, 'weight')
77     nx.draw_networkx_edge_labels(dothi, vtrirut, edge_labels=labels, ax=ax)
78     canhddnn = list(zip(ddnn, ddnn[1:]))
79     nx.draw_networkx_edges(dothi, vtrirut, edgelist=canhddnn, edge_color='r', width=2, ax=ax)
80     nx.draw_networkx_nodes(dothi, vtrirut, nodelist=[start_node], node_color='g', node_size=500, ax=ax)
81     nx.draw_networkx_nodes(dothi, vtrirut, nodelist=[end_node], node_color='b', node_size=500, ax=ax)
82     canvas = FigureCanvasTkAgg(fig, master=bientk)
83     canvas.draw()
84     canvas.get_tk_widget().pack()
85 bientk = tk.Tk()
86 bientk.title("Mô phỏng thuật toán Dijkstra")
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.6

6:47 PM

VỀ ĐỒ THỊ!



The image shows a screenshot of a Visual Studio Code editor window. The editor is open to a file named `thuattoandijkstra.py`. The code is a Python script that uses the Tkinter library to create a graphical user interface for a Dijkstra's algorithm calculator. The GUI includes labels for the number of nodes, the names of the nodes, the weight matrix, and buttons for starting and ending the calculation. The script also includes a `mainloop()` call to start the GUI.

```
66 def hienDt(dothi, start_node, end_node, ddnn):
84     canvas.get_tk_widget().pack()
85     bientk = tk.Tk()
86     bientk.title("Mô phỏng thuật toán Dijkstra")
87     sld = tk.Label(bientk, text="Số lượng đỉnh:")
88     sld.pack()
89     nhapsld = tk.Entry(bientk)
90     nhapsld.pack()
91     tendinh = tk.Label(bientk, text="Tên các đỉnh (cách nhau bởi dấu phẩy):")
92     tendinh.pack()
93     nhaptendinh = tk.Entry(bientk)
94     nhaptendinh.pack()
95     matran = tk.Label(bientk, text="Ma trận trọng số (mỗi hàng cách nhau bởi dấu xuống dòng):")
96     matran.pack()
97     nhapmt = tk.Text(bientk, height=10, width=30)
98     nhapmt.pack()
99     nutbd = tk.Label(bientk, text="Nút bắt đầu:")
100    nutbd.pack()
101    nhapnutbd = tk.Entry(bientk)
102    nhapnutbd.pack()
103    nutkt = tk.Label(bientk, text="Nút kết thúc:")
104    nutkt.pack()
105    nhapnutkt = tk.Entry(bientk)
106    nhapnutkt.pack()
107    nuttinh = tk.Button(bientk, text="Tính toán", command=tinh)
108    nuttinh.pack()
109    kq = tk.Text(bientk, height=10, width=30)
110    kq.pack()
111    bientk.mainloop()
112
```

GIAO DIỆN

The image shows a web browser window with the title "Mô phỏng thuật toán Dijkstra". The page has a light gray background. At the top, there are three input fields with labels in Vietnamese: "Số lượng đỉnh:" (Number of vertices:), "Tên các đỉnh (cách nhau bởi dấu phẩy):" (Names of vertices (separated by commas):), and "Ma trận trọng số (mỗi hàng cách nhau bởi dấu xuống dòng):" (Weighted matrix (each row separated by a line break):). Below these is a large empty rectangular box for the matrix. Further down are three more input fields labeled "Nút bắt đầu:" (Start node:), "Nút kết thúc:" (End node:), and a button labeled "Tính toán" (Calculate). Below the button is another large empty rectangular box for the output. The browser's address bar shows a local file path. The Windows taskbar is visible at the bottom with various icons and the system clock showing 6:47 PM.