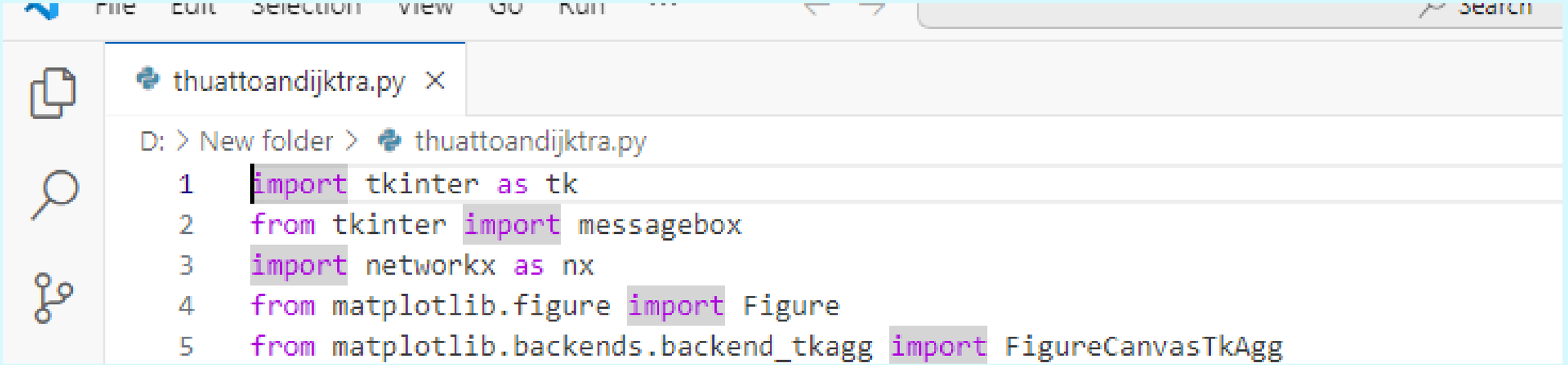
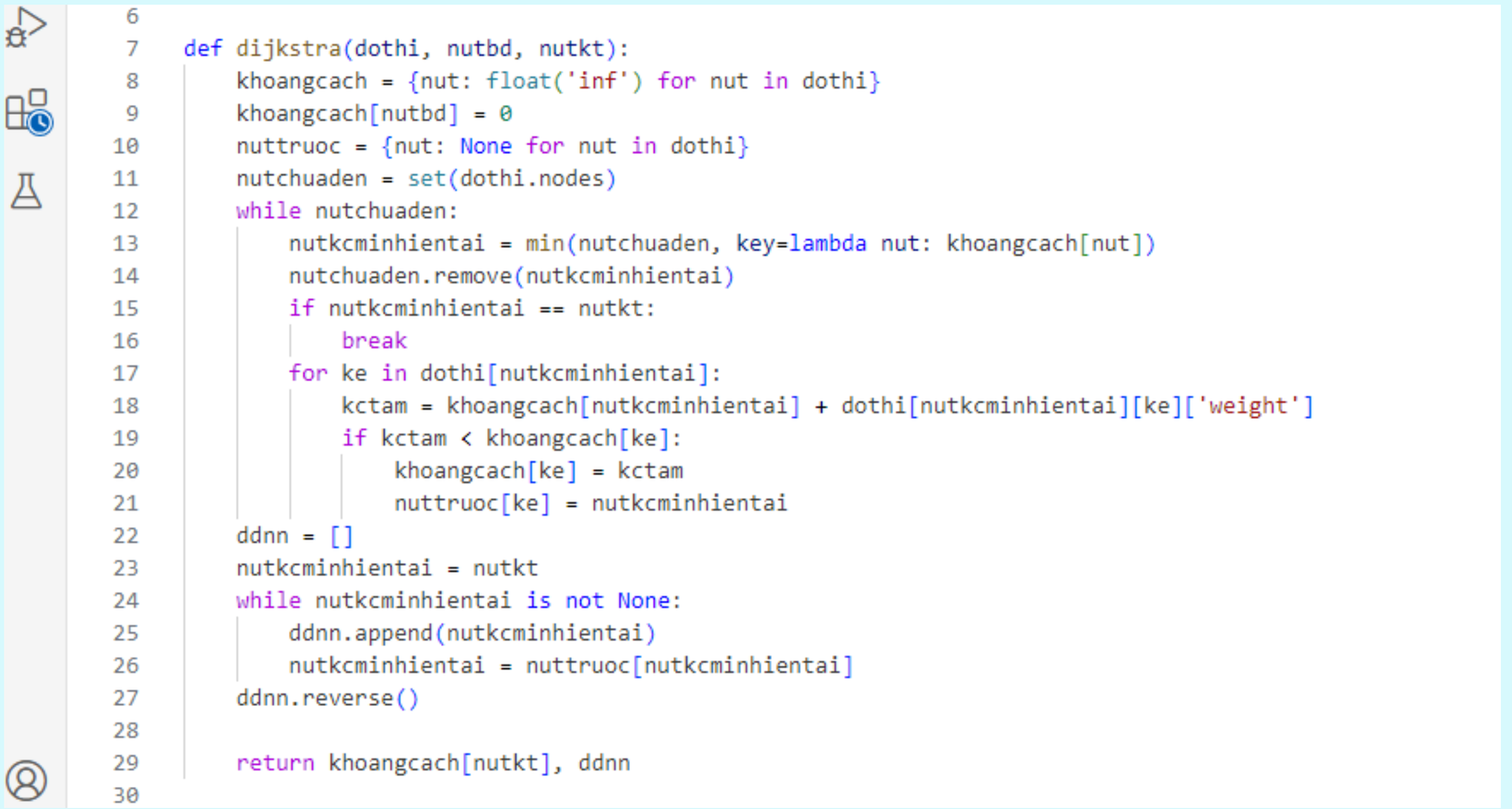
VIẾT CHƯƠNG TRÌNH MÔ PHỎNG THUẬT TOÁN DIJKTRA

THƯ VIỆN

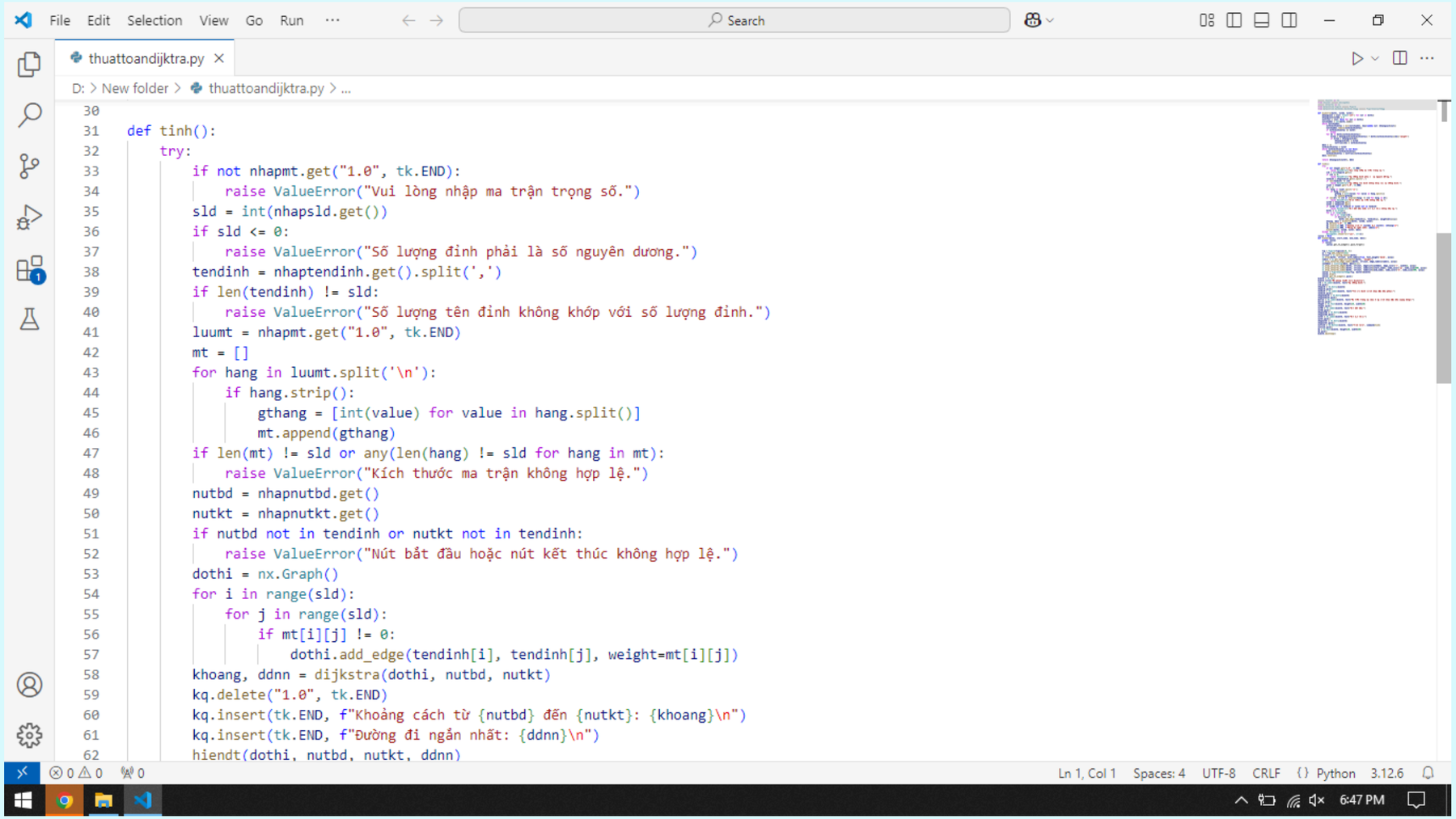

File Edit Selection View Go Run ... Search
thuattoandijktra.py
D: > New folder > thuattoandijktra.py
1 import tkinter as tk
2 from tkinter import messagebox
3 import networkx as nx
4 from matplotlib.figure import Figure
5 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

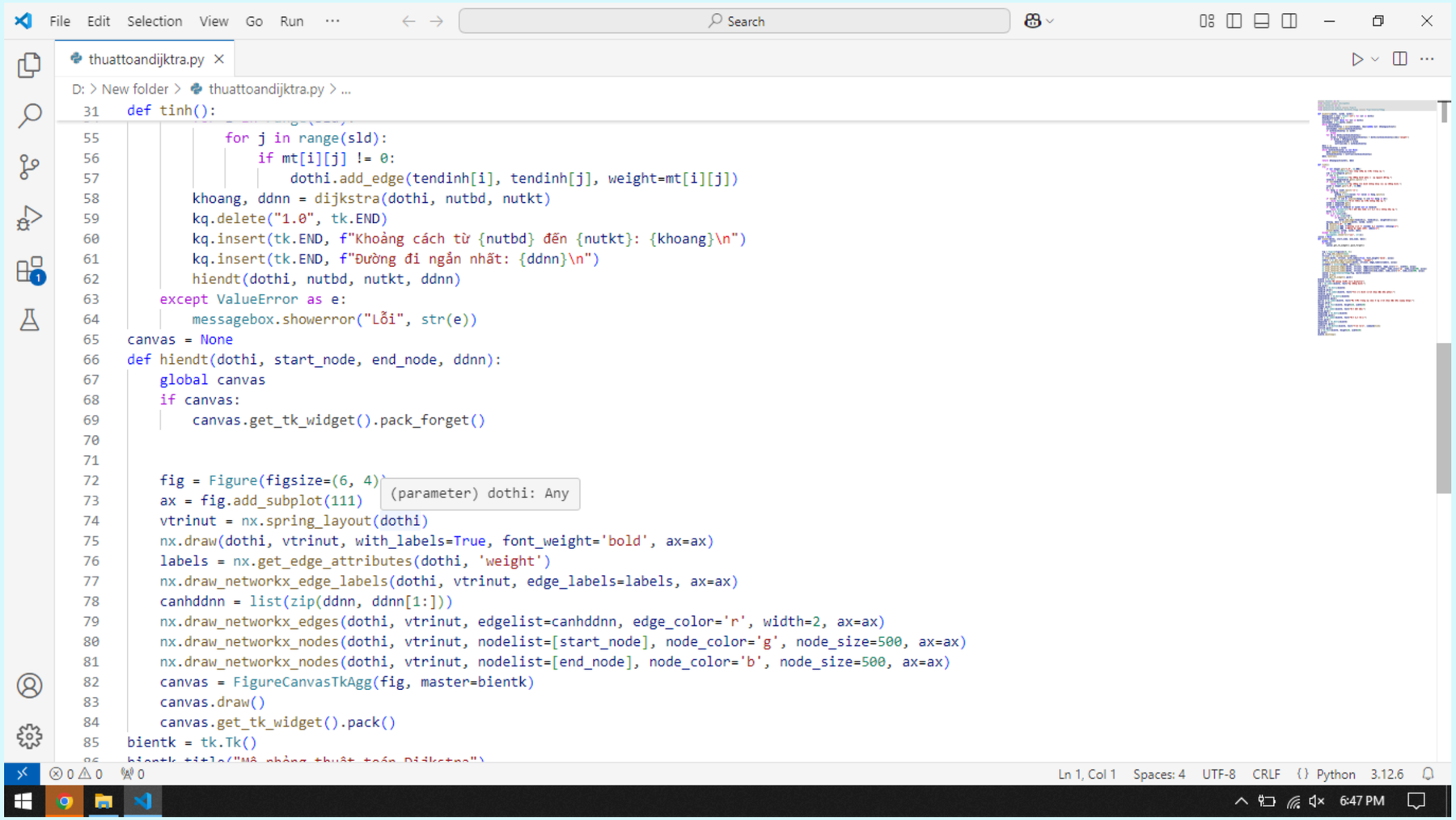
HÀM DIJKTRA



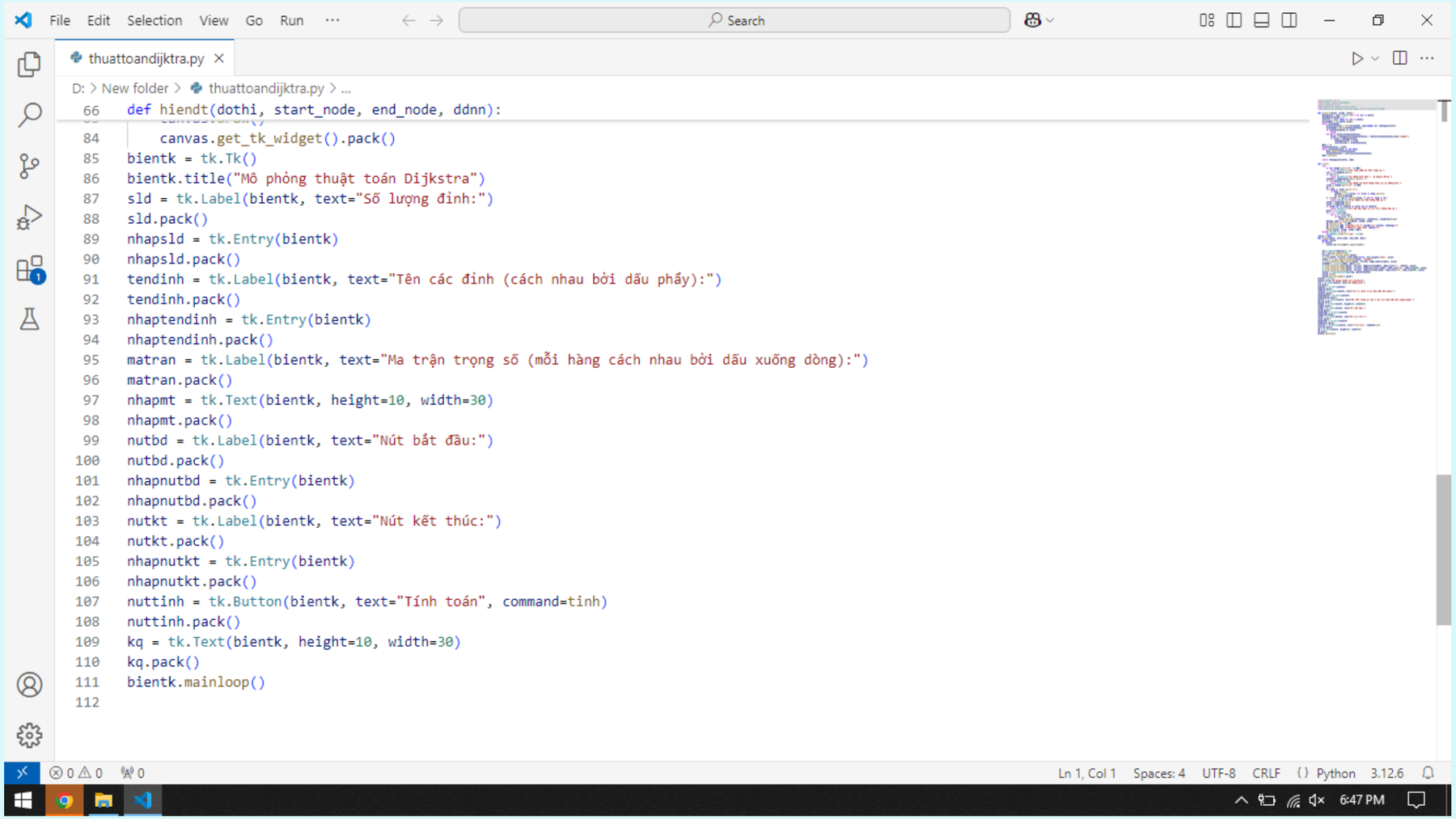
```python

def dijkstra(dothi, nutbd, nutkt):
    khoangcach = {nut: float('inf') for nut in dothi}
    khoangcach[nutbd] = 0
    nuttruoc = {nut: None for nut in dothi}
    nutchuaden = set(dothi.nodes)
    while nutchuaden:
        nutkcminhientai = min(nutchuaden, key=lambda nut: khoangcach[nut])
        nutchuaden.remove(nutkcminhientai)
        if nutkcminhientai == nutkt:
            break
        for ke in dothi[nutkcminhientai]:
            kctam = khoangcach[nutkcminhientai] + dothi[nutkcminhientai][ke]['weight']
            if kctam < khoangcach[ke]:
                khoangcach[ke] = kctam
                nuttruoc[ke] = nutkcminhientai
    ddnn = []
    nutkcminhientai = nutkt
    while nutkcminhientai is not None:
        ddnn.append(nutkcminhientai)
        nutkcminhientai = nuttruoc[nutkcminhientai]
    ddnn.reverse()

    return khoangcach[nutkt], ddnn

```

HÀM TÍNH TOÁN

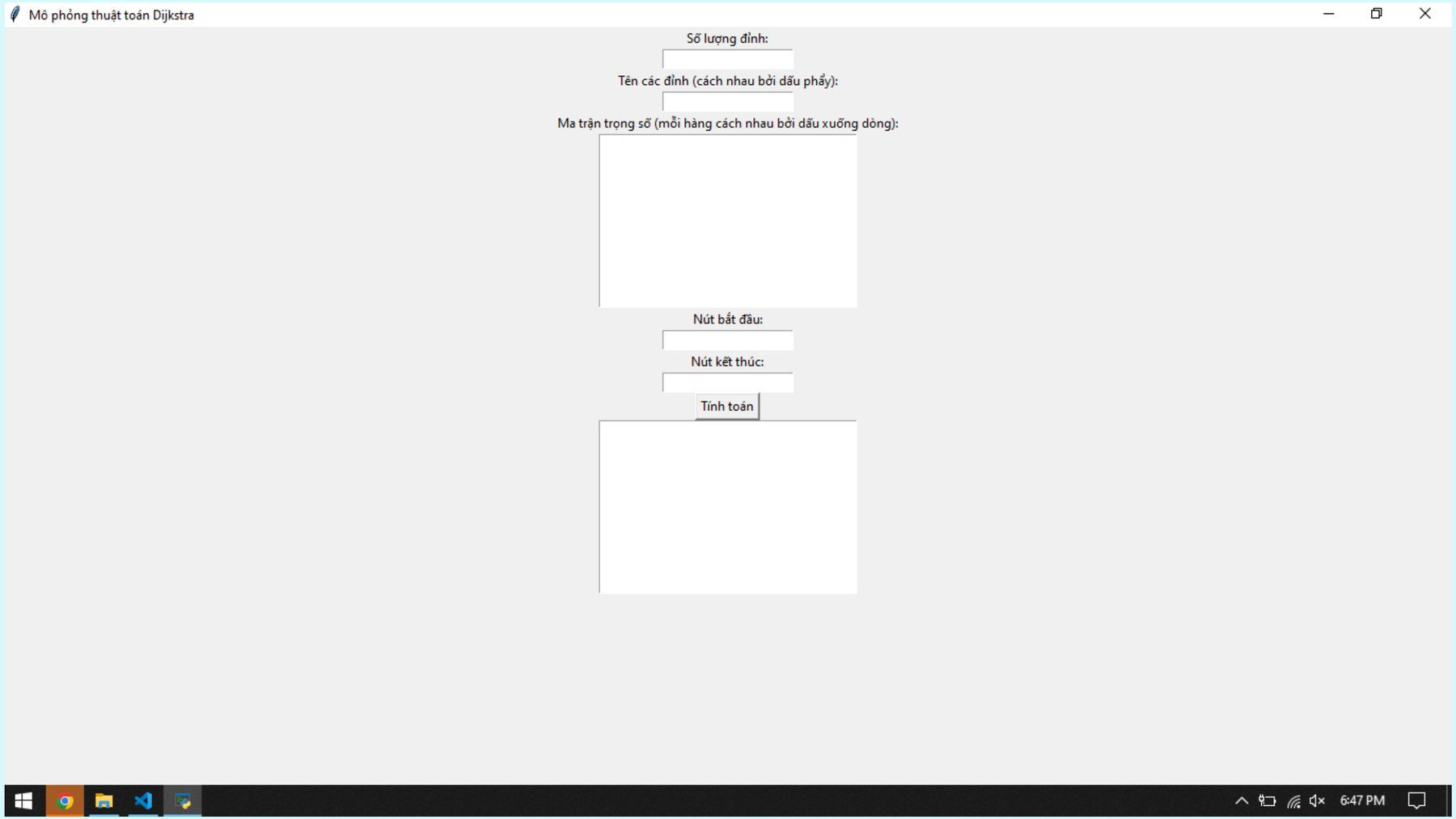



File Edit Selection View Go Run ···
Search
thuattoandijktra.py
D: > New folder > thuattoandijktra.py > ...
def tinh():
for j in range(sld):
if mt[i][j] != 0:
dothi.add_edge(tendinh[i], tendinh[j], weight=mt[i][j])
khoang, ddnn = dijkstra(dothi, nutbd, nutkt)
kq.delete("1.0", tk.END)
kq.insert(tk.END, f"Khoảng cách từ {nutbd} đến {nutkt}: {khoang}\n")
kq.insert(tk.END, f"Đường đi ngắn nhất: {ddnn}\n")
hiendt(dothi, nutbd, nutkt, ddnn)
except ValueError as e:
messagebox.showerror("Lỗi", str(e))
canvas = None
def hiendt(dothi, start_node, end_node, ddnn):
global canvas
if canvas:
canvas.get_tk_widget().pack_forget()
fig = Figure(figsize=(6, 4))
(parameter) dothi: Any
ax = fig.add_subplot(111)
vtrinut = nx.spring_layout(dothi)
nx.draw(dothi, vtrinut, with_labels=True, font_weight='bold', ax=ax)
labels = nx.get_edge_attributes(dothi, 'weight')
nx.draw_networkx_edge_labels(dothi, vtrinut, edge_labels=labels, ax=ax)
canhddnn = list(zip(ddnn, ddnn[1:]))
nx.draw_networkx_edges(dothi, vtrinut, edgelist=canhddnn, edge_color='r', width=2, ax=ax)
nx.draw_networkx_nodes(dothi, vtrinut, nodelist=[start_node], node_color='g', node_size=500, ax=ax)
nx.draw_networkx_nodes(dothi, vtrinut, nodelist=[end_node], node_color='b', node_size=500, ax=ax)
canvas = FigureCanvasTkAgg(fig, master=bientk)
canvas.draw()
canvas.get_tk_widget().pack()
bientk = tk.Tk()
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.6
6:47 PM

VẼ ĐỒ THỊ


File Edit Selection View Go Run
Search
thuattoandijktra.py
D: > New folder > thuattoandijktra.py > ...
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
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.6
6:47 PM

GIAO DIỆN


Mô phỏng thuật toán Dijkstra
Số lượng đỉnh:
Tên các đỉnh (cách nhau bởi dấu phẩy):
Ma trận trọng số (mỗi hàng cách nhau bởi dấu xuống dòng):
Nút bắt đầu:
Nút kết thúc:
Tính toán
6:47 PM