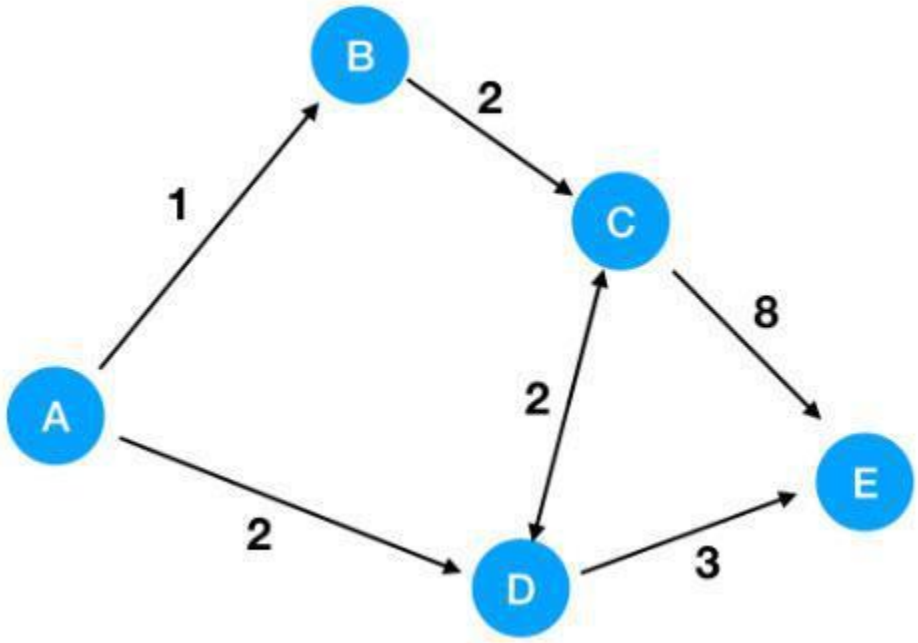
Week6q1

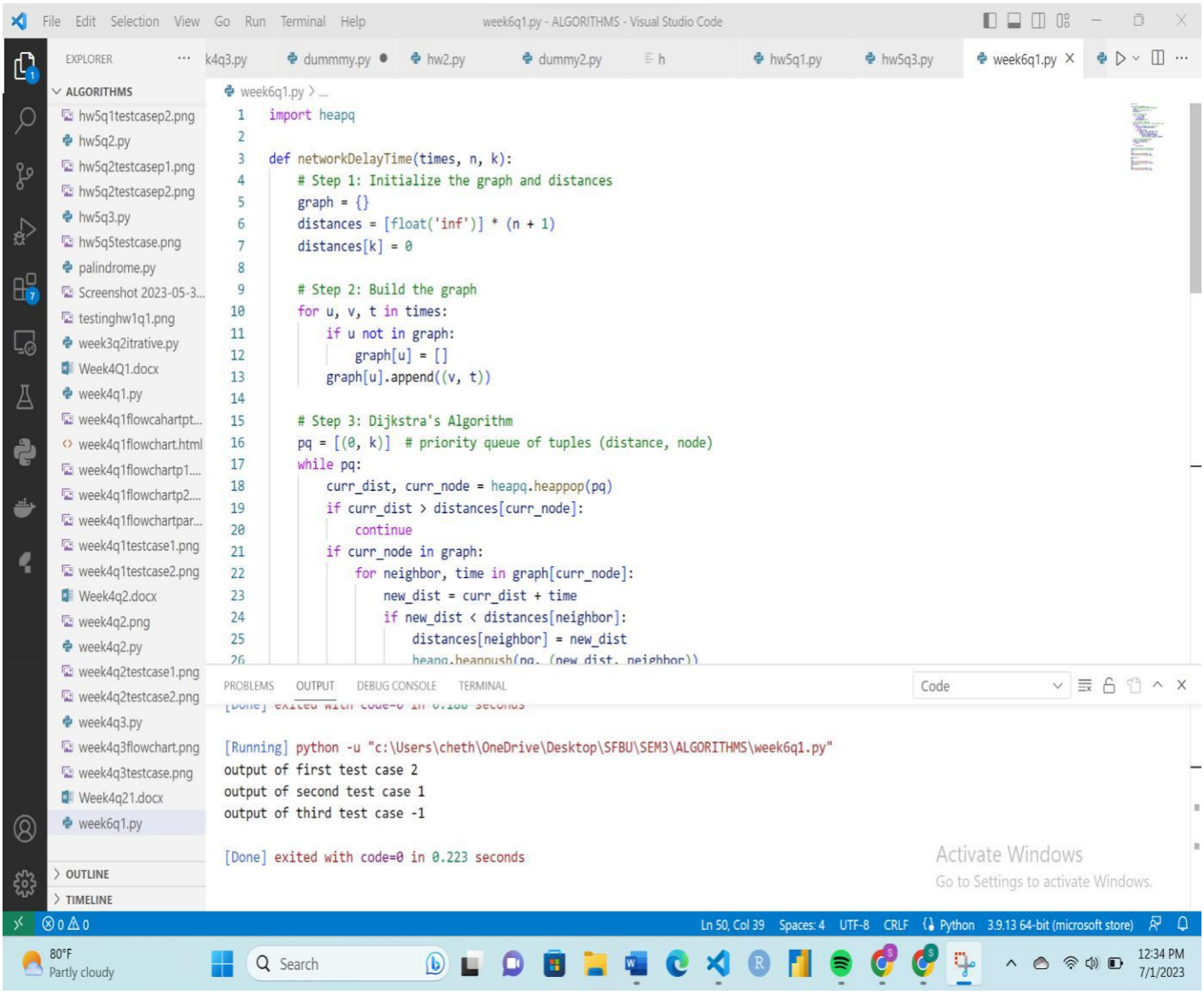


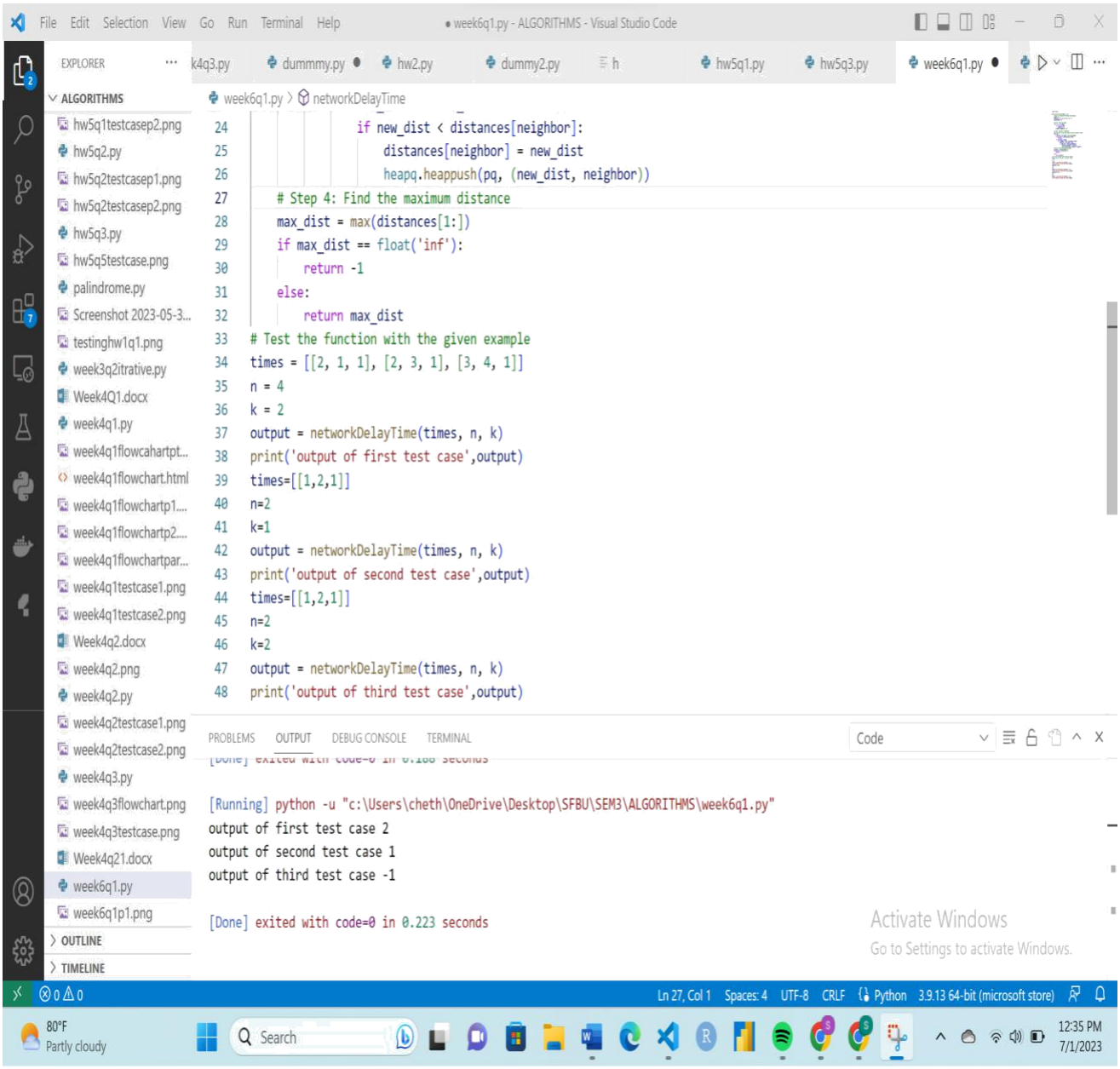
Intial	NEXT NODE A	NEXT NODE B	NEXT NODE D	NEXT NODE E	
A=∞	A=0	A=0	A=0	A=0	
B=∞	B=∞	B=1	B=1	B=1	
C=∞	C=∞	C=∞	C=3	C=3	
D=∞	D=∞	D=2	D=2	D=2	
E=∞	E=∞	E=∞	E=∞	E=5	

CHAT GPT CODE

```
import heapq

def networkDelayTime(times, n, k):
    # Step 1: Initialize the graph and distances
    graph = {}
    distances = [float('inf')] * (n + 1)
    distances[k] = 0

    # Step 2: Build the graph
    for u, v, t in times:
        if u not in graph:
            graph[u] = []
        graph[u].append((v, t))

    # Step 3: Dijkstra's Algorithm
    pq = [(0, k)]  # priority queue of tuples (distance, node)
    while pq:
        curr_dist, curr_node = heapq.heappop(pq)
        if curr_dist > distances[curr_node]:
            continue
        if curr_node in graph:
            for neighbor, time in graph[curr_node]:
                new_dist = curr_dist + time
                if new_dist < distances[neighbor]:
                    distances[neighbor] = new_dist
                    heapq.heappush(pq, (new_dist, neighbor))

    # Step 4: Find the maximum distance
    max_dist = max(distances[1:])
    if max_dist == float('inf'):
        return -1
    else:
        return max_dist

# Test the function with the given
example times = [[2, 1, 1], [2, 3, 1],
[3, 4, 1]] n = 4
k = 2
output = networkDelayTime(times, n,
k) print(output)
```

test cases


week6q1.py - ALGORITHMS - Visual Studio Code
EXPLORER
ALGORITHMS
hw5q1testcasep2.png
hw5q2.py
hw5q2testcasep1.png
hw5q2testcasep2.png
hw5q3.py
hw5q5testcase.png
palindrome.py
Screenshot 2023-05-3...
testinghw1q1.png
week3q2itrative.py
Week4Q1.docx
week4q1.py
week4q1flowcahartpt...
week4q1flowchart.html
week4q1flowchartp1....
week4q1flowchartp2....
week4q1flowchartpar...
week4q1testcase1.png
week4q1testcase2.png
Week4q2.docx
week4q2.png
week4q2.py
week4q2testcase1.png
week4q2testcase2.png
week4q3.py
week4q3flowchart.png
week4q3testcase.png
Week4q21.docx
week6q1.py
OUTLINE
TIMELINE
k4q3.py
dummmy.py
hw2.py
dummy2.py
h
hw5q1.py
hw5q3.py
week6q1.py
week6q1.py > ...
1 import heapq
2
3 def networkDelayTime(times, n, k):
4 # Step 1: Initialize the graph and distances
5 graph = {}
6 distances = [float('inf')] * (n + 1)
7 distances[k] = 0
8
9 # Step 2: Build the graph
10 for u, v, t in times:
11 if u not in graph:
12 graph[u] = []
13 graph[u].append((v, t))
14
15 # Step 3: Dijkstra's Algorithm
16 pq = [(0, k)] # priority queue of tuples (distance, node)
17 while pq:
18 curr_dist, curr_node = heapq.heappop(pq)
19 if curr_dist > distances[curr_node]:
20 continue
21 if curr_node in graph:
22 for neighbor, time in graph[curr_node]:
23 new_dist = curr_dist + time
24 if new_dist < distances[neighbor]:
25 distances[neighbor] = new_dist
26 heapq.heappush(pq, (new_dist, neighbor))
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week6q1.py"
output of first test case 2
output of second test case 1
output of third test case -1
[Done] exited with code=0 in 0.223 seconds
Activate Windows
Go to Settings to activate Windows.
Ln 50, Col 39
Spaces: 4
UTF-8
CRLF
Python
3.9.13 64-bit (microsoft store)
80°F
Partly cloudy
Search
12:34 PM
7/1/2023


File Edit Selection View Go Run Terminal Help
• week6q1.py - ALGORITHMS - Visual Studio Code
EXPLORER
ALGORITHMS
hw5q1testcasep2.png
hw5q2.py
hw5q2testcasep1.png
hw5q2testcasep2.png
hw5q3.py
hw5q5testcase.png
palindrome.py
Screenshot 2023-05-3...
testinghw1q1.png
week3q2itrative.py
Week4Q1.docx
week4q1.py
week4q1flowcahartpt...
week4q1flowchart.html
week4q1flowchartp1....
week4q1flowchartp2....
week4q1flowchartpar...
week4q1testcase1.png
week4q1testcase2.png
Week4q2.docx
week4q2.png
week4q2.py
week4q2testcase1.png
week4q2testcase2.png
week4q3.py
week4q3flowchart.png
week4q3testcase.png
Week4q21.docx
week6q1.py
week6q1p1.png
OUTLINE
TIMELINE
k4q3.py
dummmy.py
hw2.py
dummy2.py
h
hw5q1.py
hw5q3.py
week6q1.py
week6q1.py > networkDelayTime
if new_dist < distances[neighbor]:
distances[neighbor] = new_dist
heapq.heappush(pq, (new_dist, neighbor))
# Step 4: Find the maximum distance
max_dist = max(distances[1:])
if max_dist == float('inf'):
return -1
else:
return max_dist
# Test the function with the given example
times = [[2, 1, 1], [2, 3, 1], [3, 4, 1]]
n = 4
k = 2
output = networkDelayTime(times, n, k)
print('output of first test case',output)
times=[[1,2,1]]
n=2
k=1
output = networkDelayTime(times, n, k)
print('output of second test case',output)
times=[[1,2,1]]
n=2
k=2
output = networkDelayTime(times, n, k)
print('output of third test case',output)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week6q1.py"
output of first test case 2
output of second test case 1
output of third test case -1
[Done] exited with code=0 in 0.223 seconds
Activate Windows
Go to Settings to activate Windows.
Ln 27, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store)
80°F
Partly cloudy
Search
12:35 PM
7/1/2023