(1) WhatsApp | DAY-8.pdf | FREE AI Code Genera | Online Python Comp | Yogi9600/DAA | Download file | iLove

programiz.com/python-programming/online-compiler/

Programiz Python Online Compiler

Programiz PRO

main.py

Share Run

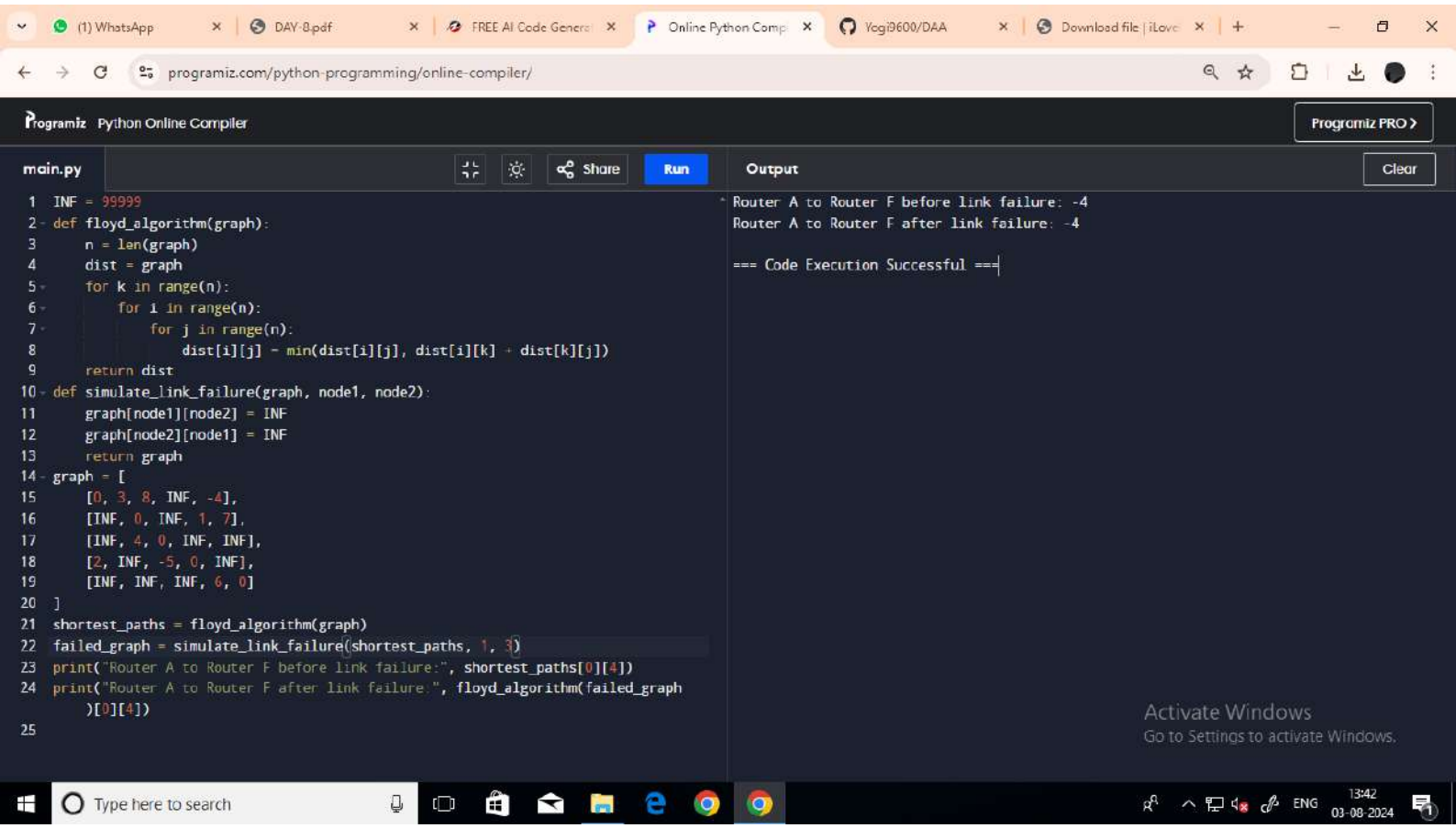
Output Clear

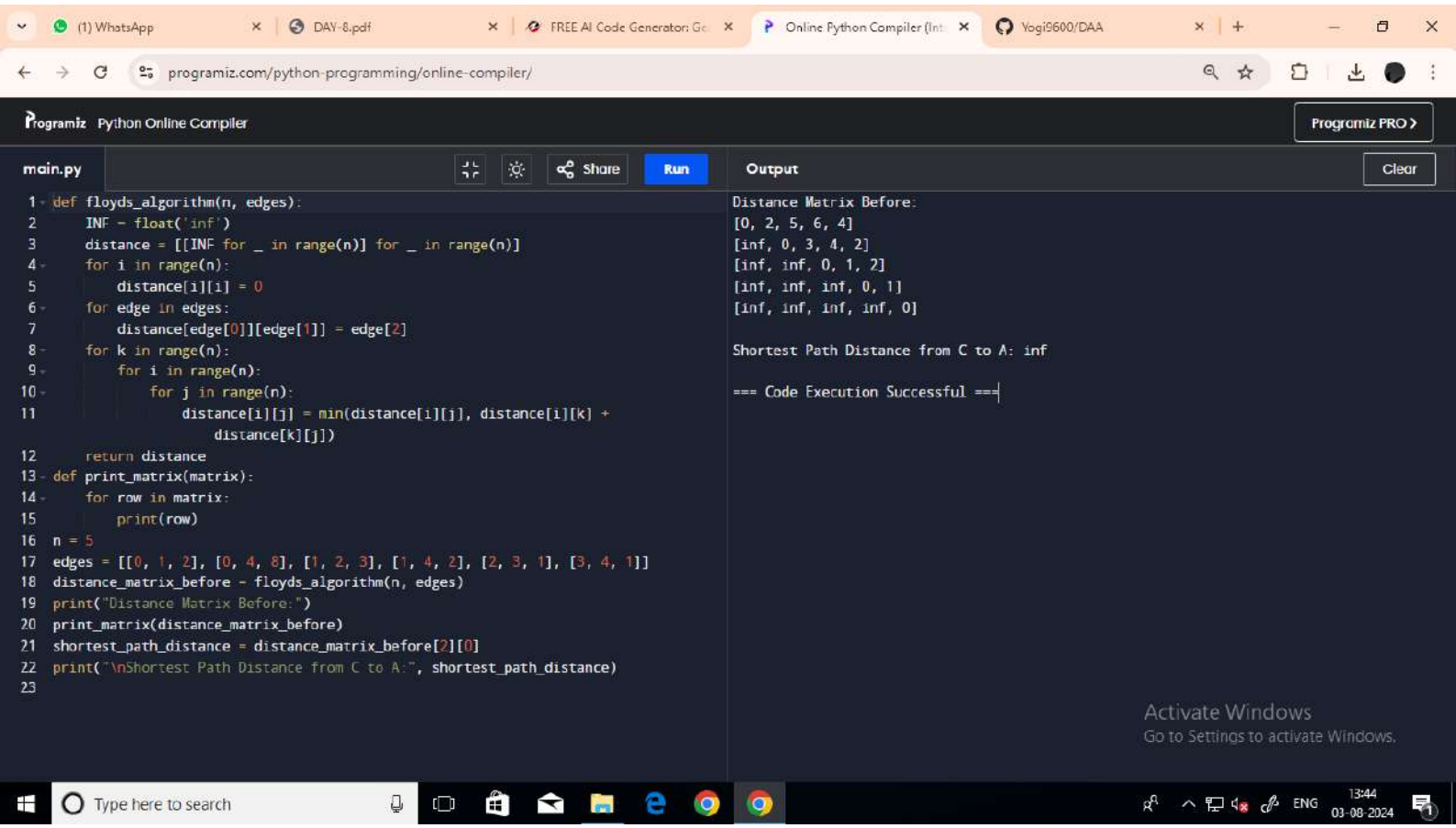
```
import sys
def floyds_algorithm(n, edges):
    distance = [[sys.maxsize for _ in range(n)] for _ in range(n)]
    for i in range(n):
        distance[i][i] = 0
    for edge in edges:
        distance[edge[0]][edge[1]] = edge[2]
    for k in range(n):
        for i in range(n):
            for j in range(n):
                distance[i][j] = min(distance[i][j], distance[i][k] +
                    distance[k][j])
    return distance
n = 4
edges = [[0, 1, 3], [1, 2, 1], [1, 3, 4], [2, 3, 1]]
distance_matrix = floyds_algorithm(n, edges)
print("Distance Matrix:")
for row in distance_matrix:
    print(row)
shortest_path = distance_matrix[1][3]
print(f"Shortest Path from City 1 to City 3: {shortest_path}")

```

```
A module you have imported isn't available at the moment. It will be available
    soon.
```


Type here to search

ENG 13:38 03-08-2024


(1) WhatsApp
DAY-8.pdf
FREE AI Code Genera
Online Python Comp
Yogi9600/DAA
Download file | iLove
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
INF = 99999
def floyd_algorithm(graph):
n = len(graph)
dist = graph
for k in range(n):
for i in range(n):
for j in range(n):
dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
return dist
def simulate_link_failure(graph, node1, node2):
graph[node1][node2] = INF
graph[node2][node1] = INF
return graph
graph = [
[0, 3, 8, INF, -4],
[INF, 0, INF, 1, 7],
[INF, 4, 0, INF, INF],
[2, INF, -5, 0, INF],
[INF, INF, INF, 6, 0]
]
shortest_paths = floyd_algorithm(graph)
failed_graph = simulate_link_failure(shortest_paths, 1, 3)
print("Router A to Router F before link failure:", shortest_paths[0][4])
print("Router A to Router F after link failure:", floyd_algorithm(failed_graph
)[0][4])
Router A to Router F before link failure: -4
Router A to Router F after link failure: -4
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
13:42
03-08-2024


(1) WhatsApp
DAY-8.pdf
FREE AI Code Generator: Ge
Online Python Compiler (Int
Yogi9600/DAA
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
def floyds_algorithm(n, edges):
INF = float('inf')
distance = [[INF for _ in range(n)] for _ in range(n)]
for i in range(n):
distance[i][i] = 0
for edge in edges:
distance[edge[0]][edge[1]] = edge[2]
for k in range(n):
for i in range(n):
for j in range(n):
distance[i][j] = min(distance[i][j], distance[i][k] +
distance[k][j])
return distance
def print_matrix(matrix):
for row in matrix:
print(row)
n = 5
edges = [[0, 1, 2], [0, 4, 8], [1, 2, 3], [1, 4, 2], [2, 3, 1], [3, 4, 1]]
distance_matrix_before = floyds_algorithm(n, edges)
print("Distance Matrix Before:")
print_matrix(distance_matrix_before)
shortest_path_distance = distance_matrix_before[2][0]
print("\nShortest Path Distance from C to A:", shortest_path_distance)
Distance Matrix Before:
[0, 2, 5, 6, 4]
[inf, 0, 3, 4, 2]
[inf, inf, 0, 1, 2]
[inf, inf, inf, 0, 1]
[inf, inf, inf, inf, 0]
Shortest Path Distance from C to A: inf
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
13:44
03-08-2024

(2) WhatsApp | DAY-8.pdf | FREE AI Code Generator: Ge | Online Python Compiler (Int | Yogi9600/DAA

programiz.com/python-programming/online-compiler/

Programiz Python Online Compiler

Programiz PRO >

main.py

Share | Run

```
def optimal_bst(keys, freq):
    n = len(keys)
    cost = [[0 for _ in range(n)] for _ in range(n)]
    root = [[0 for _ in range(n)] for _ in range(n)]
    for i in range(n):
        cost[i][i] = freq[i]
        root[i][i] = i
    for L in range(2, n + 1):
        for i in range(n - L + 1):
            j = i + L - 1
            cost[i][j] = float('inf')
            for r in range(i, j + 1):
                c = cost[i][r - 1] if r > i else 0
                c += cost[r + 1][j] if r < j else 0
                c += sum(freq[i:j + 1])
                if c < cost[i][j]:
                    cost[i][j] = c
                    root[i][j] = r
    return cost[0][n - 1], cost, root
keys = ['A', 'B', 'C', 'D']
freq = [0.1, 0.2, 0.4, 0.3]
cost, cost_table, root_table = optimal_bst(keys, freq)
print("Cost:", cost)
print("Cost Table:")
for row in cost_table:
    print(row)
print("Root Table:")
```

Output | Clear

```
Cost: 1.7
Cost Table:
[0.1, 0.4, 1.1, 1.7]
[0, 0.2, 0.8, 1.4000000000000001]
[0, 0, 0.4, 1.0]
[0, 0, 0, 0.3]
Root Table:
[0, 1, 2, 2]
[0, 1, 2, 2]
[0, 0, 2, 2]
[0, 0, 0, 3]

=== Code Execution Successful ===
```


Type here to search

ENG 13:55 03-08-2024

(2) WhatsApp | DAV-8.pdf | FREE AI Code Generator: Ge | Online Python Compiler (Int | Yogi9600/DAA

programiz.com/python-programming/online-compiler/

Programiz Python Online Compiler

Programiz PRO >

main.py

Share | Run

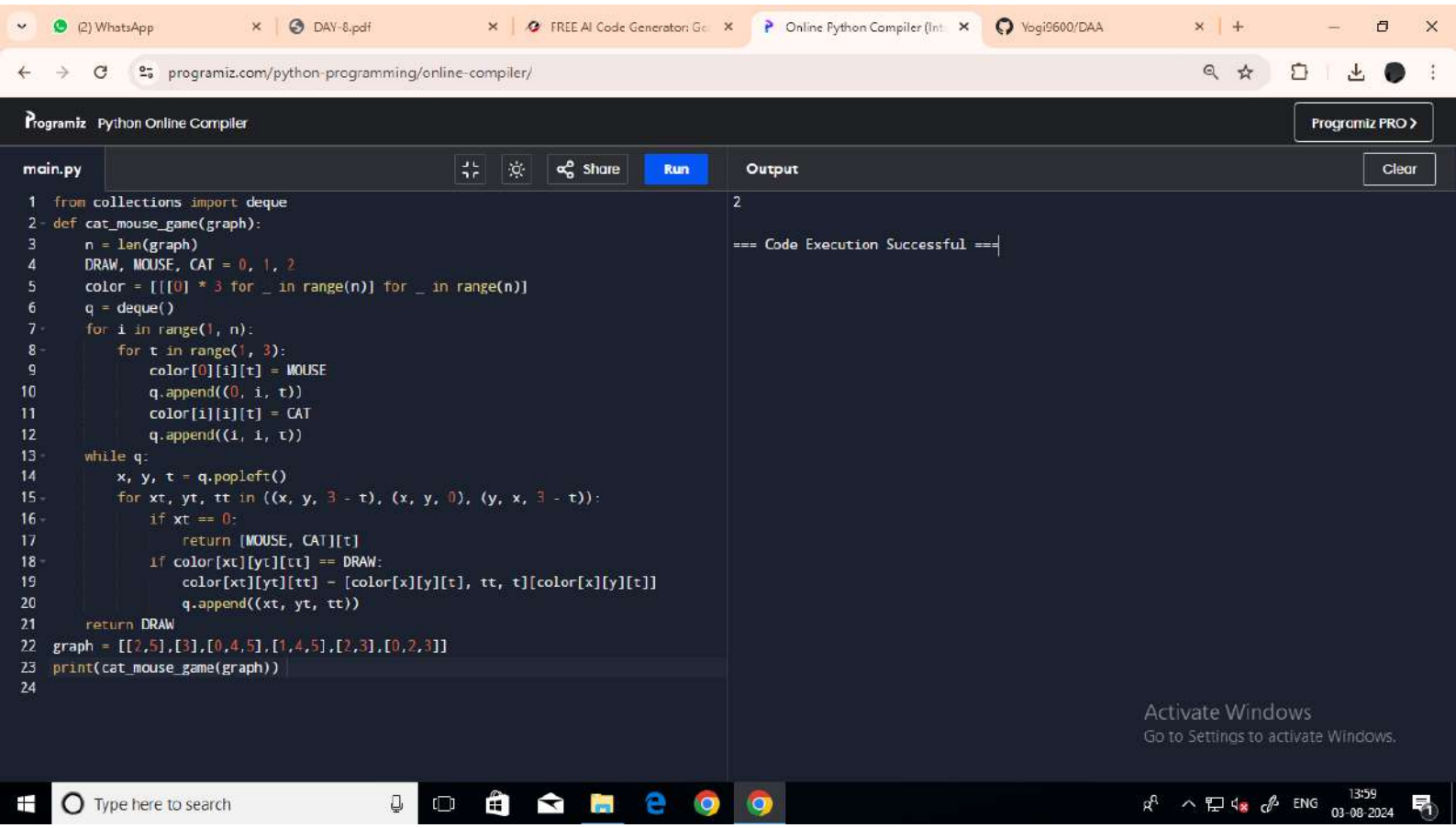
```python
def optimal_bst(keys, freq):
    n = len(keys)
    cost = [[0 for _ in range(n)] for _ in range(n)]
    root = [[0 for _ in range(n)] for _ in range(n)]
    for i in range(n):
        cost[i][i] = freq[i]
        root[i][i] = i
    for L in range(2, n + 1):
        for i in range(n - L + 1):
            j = i + L - 1
            cost[i][j] = float('inf')
            for r in range(i, j + 1):
                c = cost[i][r - 1] if r > i else 0
                c += cost[r + 1][j] if r < j else 0
                c += sum(freq[i:j + 1])
                if c < cost[i][j]:
                    cost[i][j] = c
                    root[i][j] = r
    return cost[0][n - 1], root
keys1 = [10, 12]
freq1 = [34, 50]
keys2 = [10, 12, 20]
freq2 = [34, 8, 50]
result1, _ = optimal_bst(keys1, freq1)
result2, _ = optimal_bst(keys2, freq2)
print("Output for Test Case 1:", result1)
print("Output for Test Case 2:", result2)

```

Output | Clear

```
Output for Test Case 1: 118
Output for Test Case 2: 142

=== Code Execution Successful ===
```


Type here to search

ENG 13:56 03-08-2024


(2) WhatsApp
DAY-8.pdf
FREE AI Code Generator: Ge
Online Python Compiler (Int
Yogi9600/DAA
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
from collections import deque
def cat_mouse_game(graph):
n = len(graph)
DRAW, MOUSE, CAT = 0, 1, 2
color = [[[0] * 3 for _ in range(n)] for _ in range(n)]
q = deque()
for i in range(1, n):
for t in range(1, 3):
color[0][i][t] = MOUSE
q.append((0, i, t))
color[i][i][t] = CAT
q.append((i, i, t))
while q:
x, y, t = q.popleft()
for xt, yt, tt in ((x, y, 3 - t), (x, y, 0), (y, x, 3 - t)):
if xt == 0:
return [MOUSE, CAT][t]
if color[xt][yt][tt] == DRAW:
color[xt][yt][tt] = [color[x][y][t], tt, t][color[x][y][t]]
q.append((xt, yt, tt))
return DRAW
graph = [[2,5],[3],[0,4,5],[1,4,5],[2,3],[0,2,3]]
print(cat_mouse_game(graph))
2
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
13:59
03-08-2024

(2) WhatsApp | DAY-8.pdf | FREE AI Code Generator: Ge | Online Python Compiler (Int | Yogi9600/DAA

programiz.com/python-programming/online-compiler/

Programiz Python Online Compiler

Programiz PRO >

main.py

Share Run

Output Clear

```python
from collections import defaultdict
import heapq
def maxProbability(n, edges, succProb, start, end):
    graph = defaultdict(list)
    for i, (a, b) in enumerate(edges):
        graph[a].append((b, succProb[i]))
        graph[b].append((a, succProb[i]))
    pq = [(-1, start)]
    probs = [0] * n
    probs[start] = 1
    while pq:
        prob, node = heapq.heappop(pq)
        if node == end:
            return -prob
        for nei, nei_prob in graph[node]:
            if probs[nei] < prob * nei_prob:
                probs[nei] = prob * nei_prob
                heapq.heappush(pq, (-probs[nei], nei))
    return 0
```

```
=== Code Execution Successful ===
```


Type here to search

ENG 14:00 03-08-2024

(2) WhatsApp | DAY-8.pdf | FREE AI Code Generator: Ge | Online Python Compiler (Int | Yogi9600/DAA

programiz.com/python-programming/online-compiler/

Programiz Python Online Compiler

Programiz PRO >

main.py

Share | Run

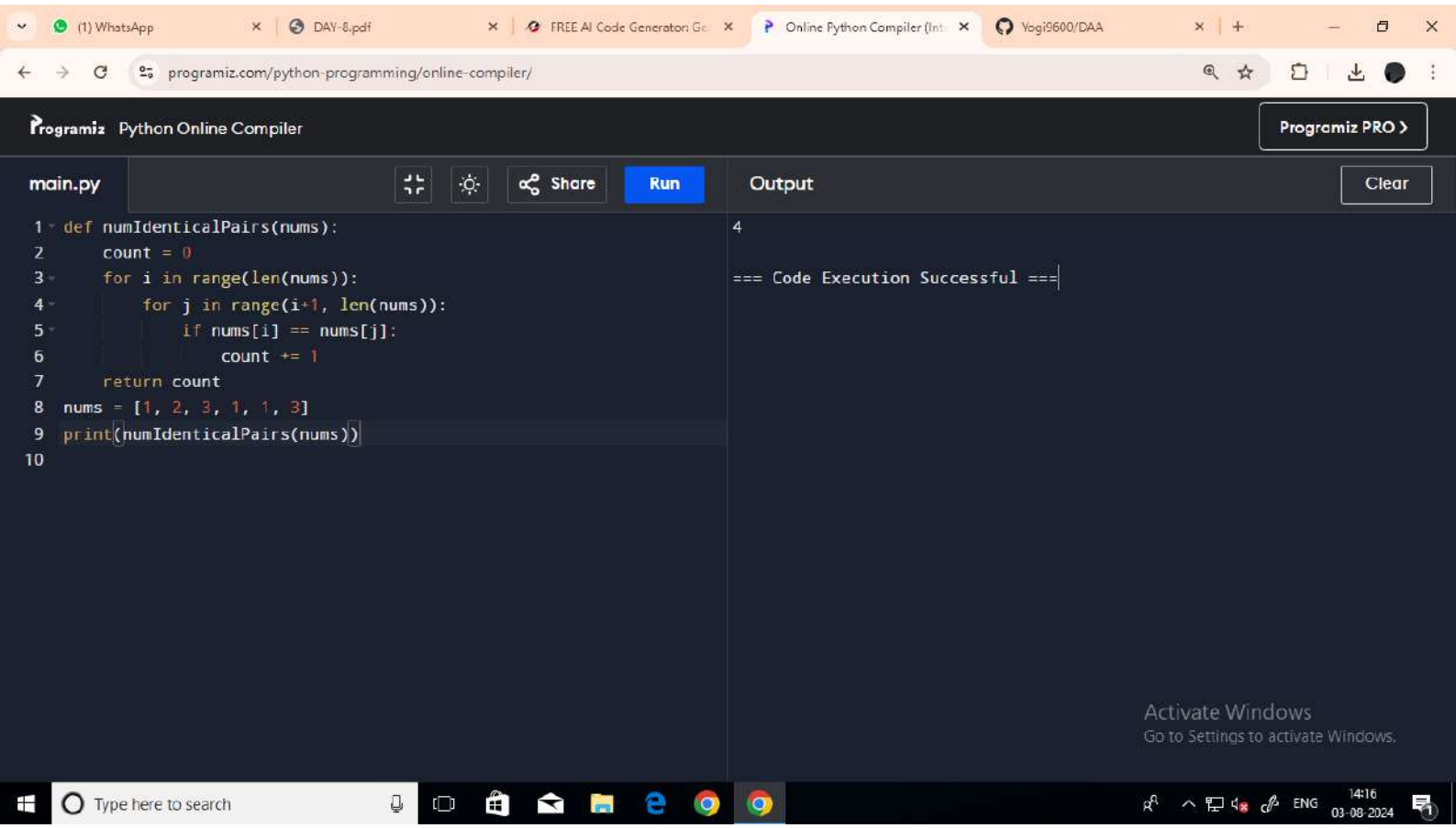
```
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        dp = [[1] * n for _ in range(m)]
        for i in range(1, m):
            for j in range(1, n):
                dp[i][j] = dp[i - 1][j] + dp[i][j - 1]
        return dp[m - 1][n - 1]
sol = Solution()
m = 3
n = 7
output = sol.uniquePaths(m, n)
print(output)

```

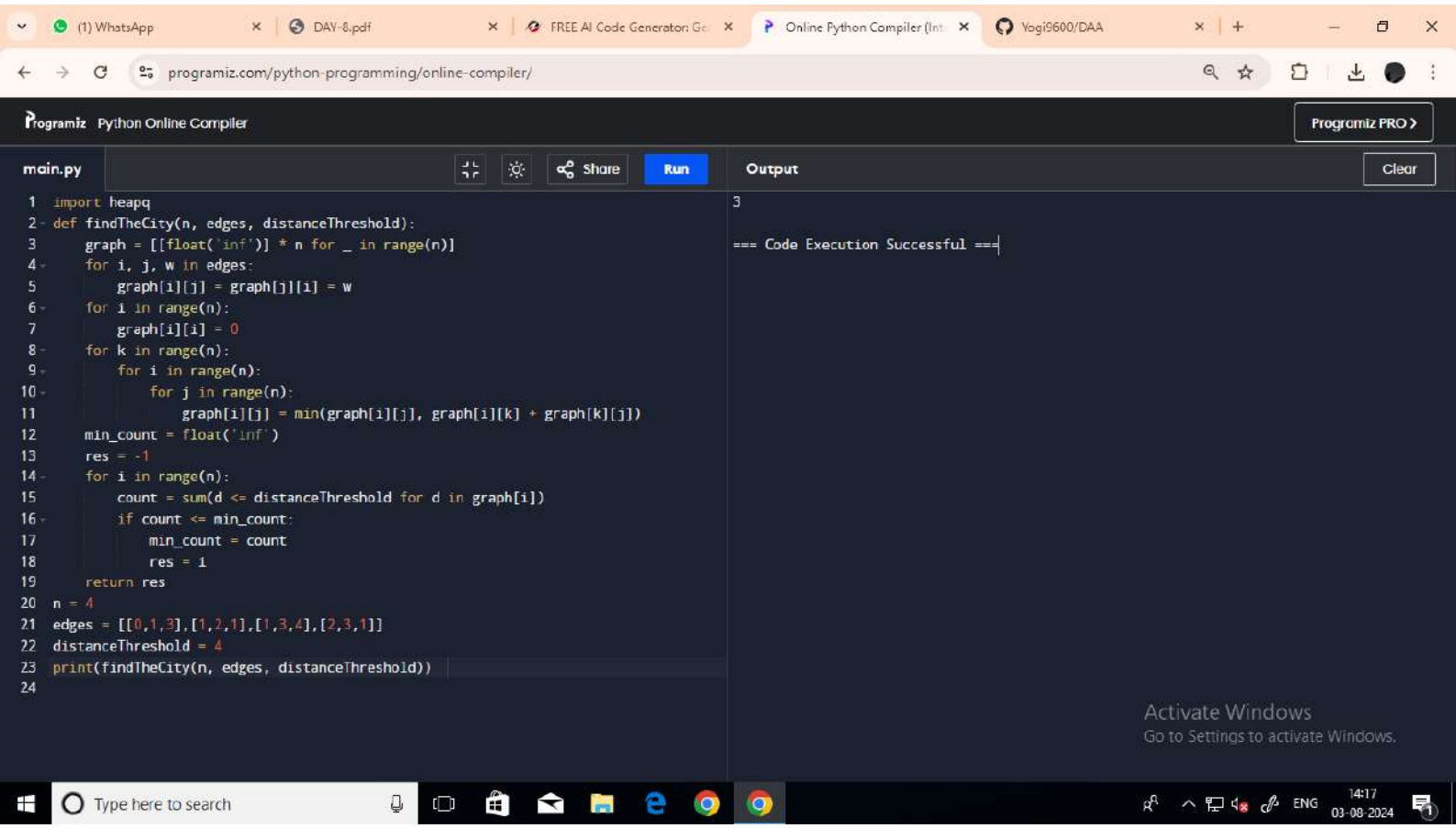
Output | Clear

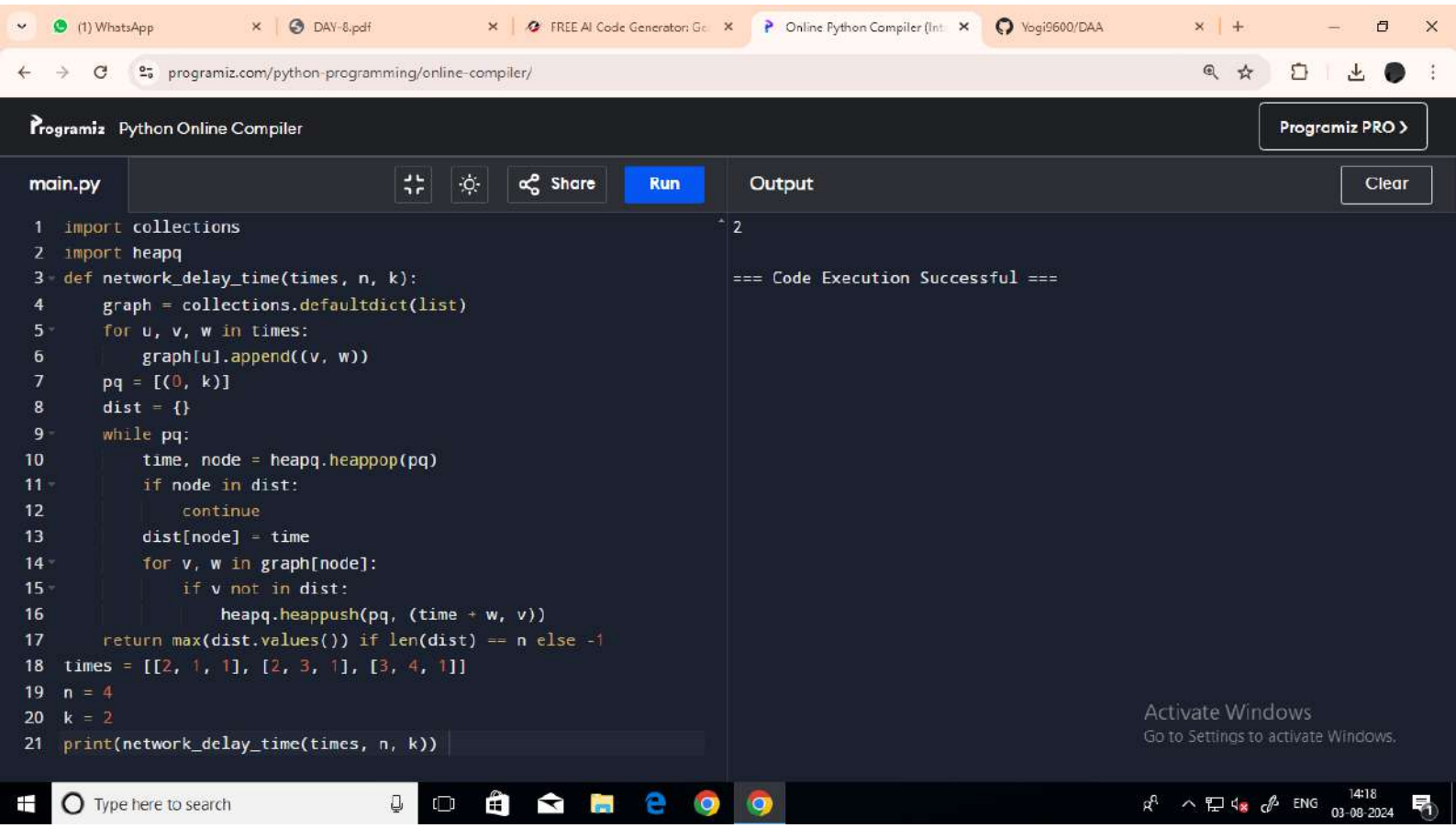
```
28

=== Code Execution Successful ===
```


Type here to search

ENG 14:02 03-08-2024


(1) WhatsApp
DAY-8.pdf
FREE AI Code Generator: Ge
Online Python Compiler (Int
Yogi9600/DAA
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
def numIdenticalPairs(nums):
count = 0
for i in range(len(nums)):
for j in range(i+1, len(nums)):
if nums[i] == nums[j]:
count += 1
return count
nums = [1, 2, 3, 1, 1, 3]
print(numIdenticalPairs(nums))
4
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
14:16
03-08-2024


(1) WhatsApp
DAY-8.pdf
FREE AI Code Generator: Ge
Online Python Compiler (Int
Yogi9600/DAA
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
import heapq
def findTheCity(n, edges, distanceThreshold):
graph = [[float('inf')] * n for _ in range(n)]
for i, j, w in edges:
graph[i][j] = graph[j][i] = w
for i in range(n):
graph[i][i] = 0
for k in range(n):
for i in range(n):
for j in range(n):
graph[i][j] = min(graph[i][j], graph[i][k] + graph[k][j])
min_count = float('inf')
res = -1
for i in range(n):
count = sum(d <= distanceThreshold for d in graph[i])
if count <= min_count:
min_count = count
res = i
return res
n = 4
edges = [[0,1,3],[1,2,1],[1,3,4],[2,3,1]]
distanceThreshold = 4
print(findTheCity(n, edges, distanceThreshold))
3
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
14:17
03-08-2024


(1) WhatsApp
DAY-8.pdf
FREE AI Code Generator: Ge
Online Python Compiler (Int
Yogi9600/DAA
programiz.com/python-programming/online-compiler/
Programiz Python Online Compiler
Programiz PRO
main.py
Share
Run
Output
Clear
import collections
import heapq
def network_delay_time(times, n, k):
graph = collections.defaultdict(list)
for u, v, w in times:
graph[u].append((v, w))
pq = [(0, k)]
dist = {}
while pq:
time, node = heapq.heappop(pq)
if node in dist:
continue
dist[node] = time
for v, w in graph[node]:
if v not in dist:
heapq.heappush(pq, (time + w, v))
return max(dist.values()) if len(dist) == n else -1
times = [[2, 1, 1], [2, 3, 1], [3, 4, 1]]
n = 4
k = 2
print(network_delay_time(times, n, k))
2
=== Code Execution Successful ===
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
14:18
03-08-2024