

Output of problem_1 of assignment_1:

1. Terminal output of comparison between memory ,time and search space of GBFS, A* ,weighted A*:

```
user@user-B560M-D53H-AC:~/Desktop/AI/problem_1_39_44$ python3 problem1.py
982
/home/user/Desktop/AI/problem_1_39_44/problem1.py:21: FutureWarning: The 'euclidean_dist_vec' function has been renamed 'euclidean'. Calling 'euclidean_dist_vec' will raise an error starting in the v2.0.0 release.
  hstd_value = ox.distance.euclidean_dist_vec(graph.nodes[node1]['y'], graph.nodes[node1]['x'],
-----
| Algorithm | Memory (Bytes) | Time (ms) | Search Space |
-----+-----+-----+-----+
| Greedy Best First Search | 80 | 0.433683 | 10 |
| A* | 432 | 2.45142 | 54 |
| Weighted A* | 5400 | 2.49767 | 54 |
-----
QSocketNotifier: Can only be used with threads started with QThread
qt.qpa.wayland: Wayland does not support QWindow::requestActivate()
qt.qpa.wayland: Wayland does not support QWindow::requestActivate()
qt.qpa.wayland: Wayland does not support QWindow::requestActivate()
qt.qpa.wayland: Wayland does not support QWindow::requestActivate()
user@user-B560M-D53H-AC:~/Desktop/AI/problem_1_39_44$
```

The screenshot shows the Visual Studio Code editor with the file `problem1.py` open. The code defines a `dijkstra` function and an `evaluation_function` for A* search. The terminal output at the bottom matches the one shown in the first block, displaying the execution results for Greedy Best First Search, A*, and Weighted A* algorithms.

```
problem1.py - AI - Visual Studio Code
19 def dijkstra(node1, node2):
20     # task_value = task_values[task_algorithm]
21
22     # # Increment the risk iterator and loop back to the beginning if necessary
23     # risk_iterator = (risk_iterator + 1) % len(risk_values)
24     return retValue
25
26 def dijkstra(graph, start_node):
27     dist = {node: float('inf') for node in graph.nodes}
28     dist[start_node] = 0
29     predecessors = {}
30     pq = [(0, start_node)]
31     while pq:
32         current_dist, current_node = heapq.heappop(pq)
33         if current_dist > dist[current_node]:
34             continue
35         for neighbor, edge in graph[current_node].items():
36             weight = edge.get('weight', 1)
37             new_dist = dist[current_node] + weight
38             if new_dist < dist[neighbor]:
39                 dist[neighbor] = new_dist
40                 predecessors[neighbor] = current_node
41                 heapq.heappush(pq, (new_dist, neighbor))
42     return dist, predecessors
43
44 def evaluation_function(node, target_node, actual_cost, heuristic, weight):
45     return actual_cost[node] + weight * heuristic(node, target_node)
46
47 def greedy_best_first_search(graph, start_node, target_node, heuristic):
```

2.Main map and hit maps:

2.1 Main map:

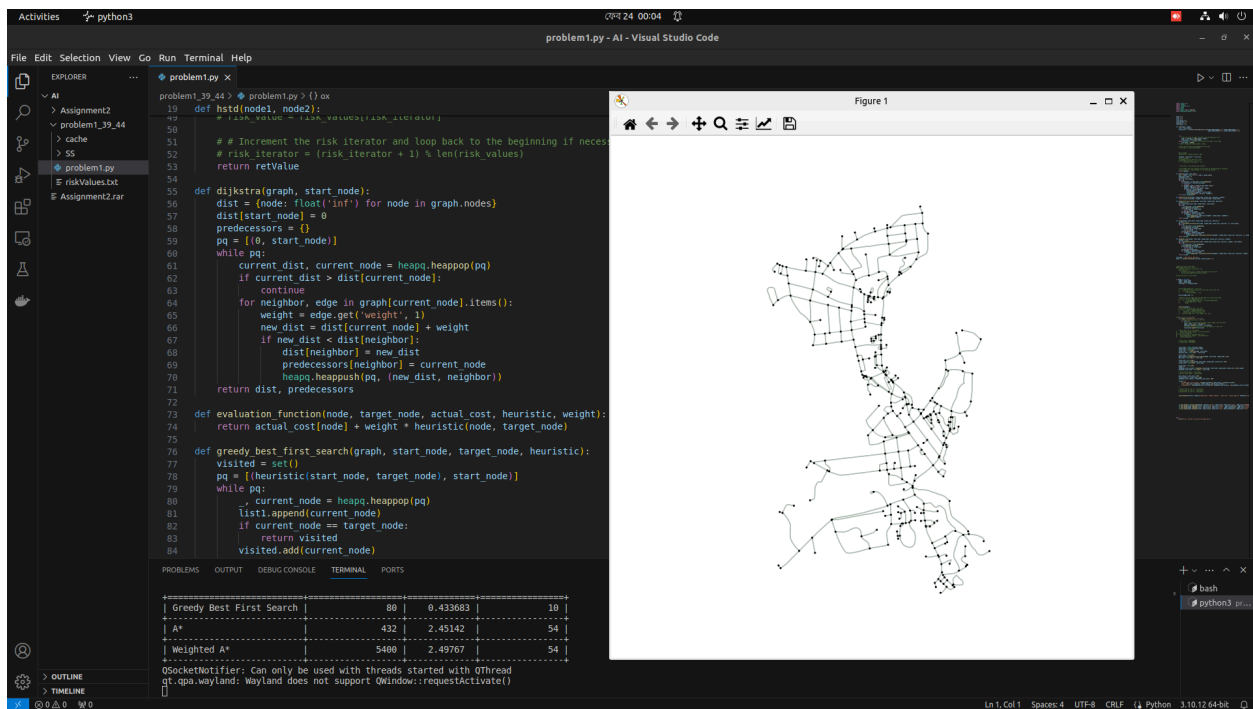


Fig:Main Map

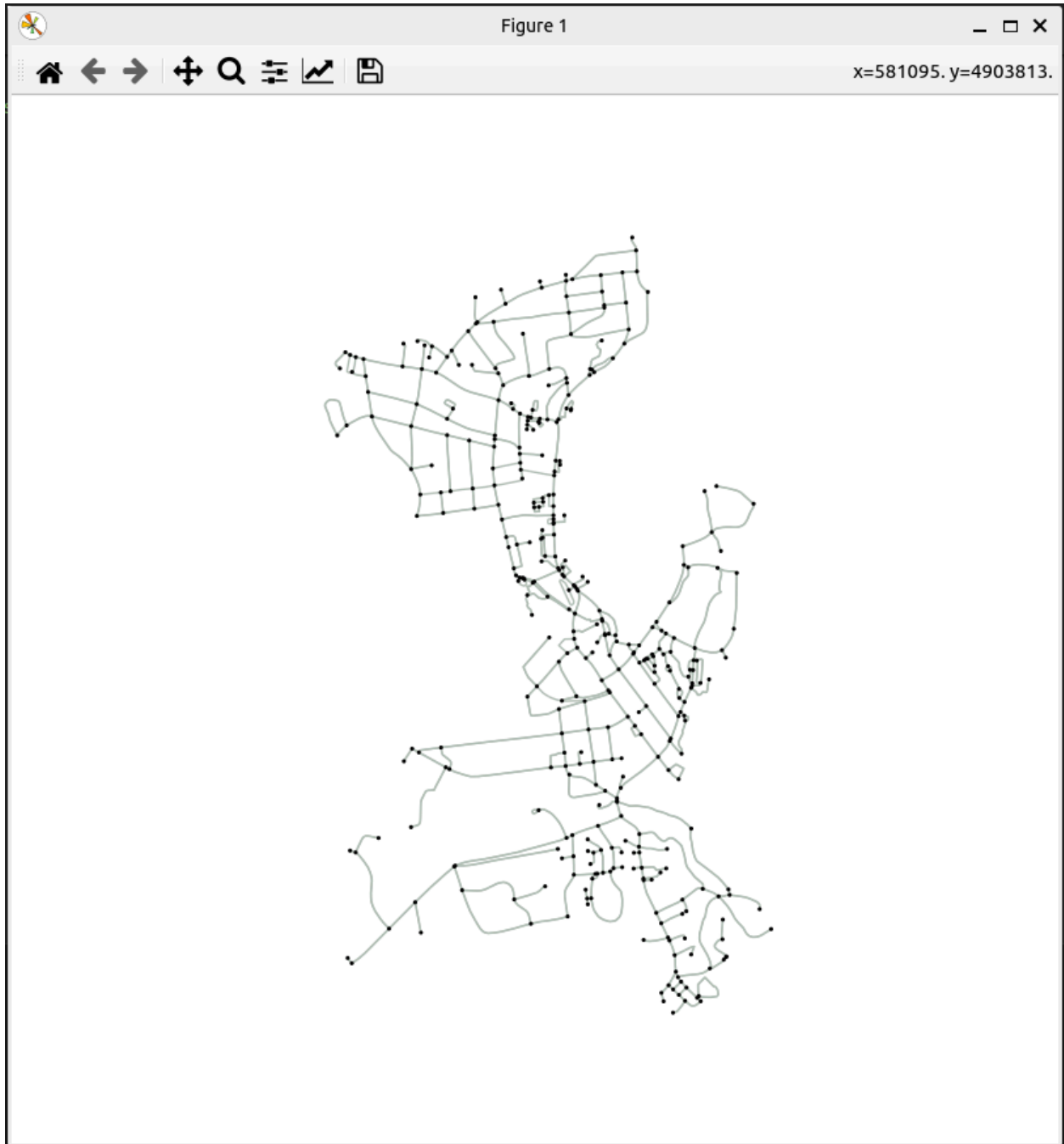


Fig:Main Map Cropped

2.2 GBFS :

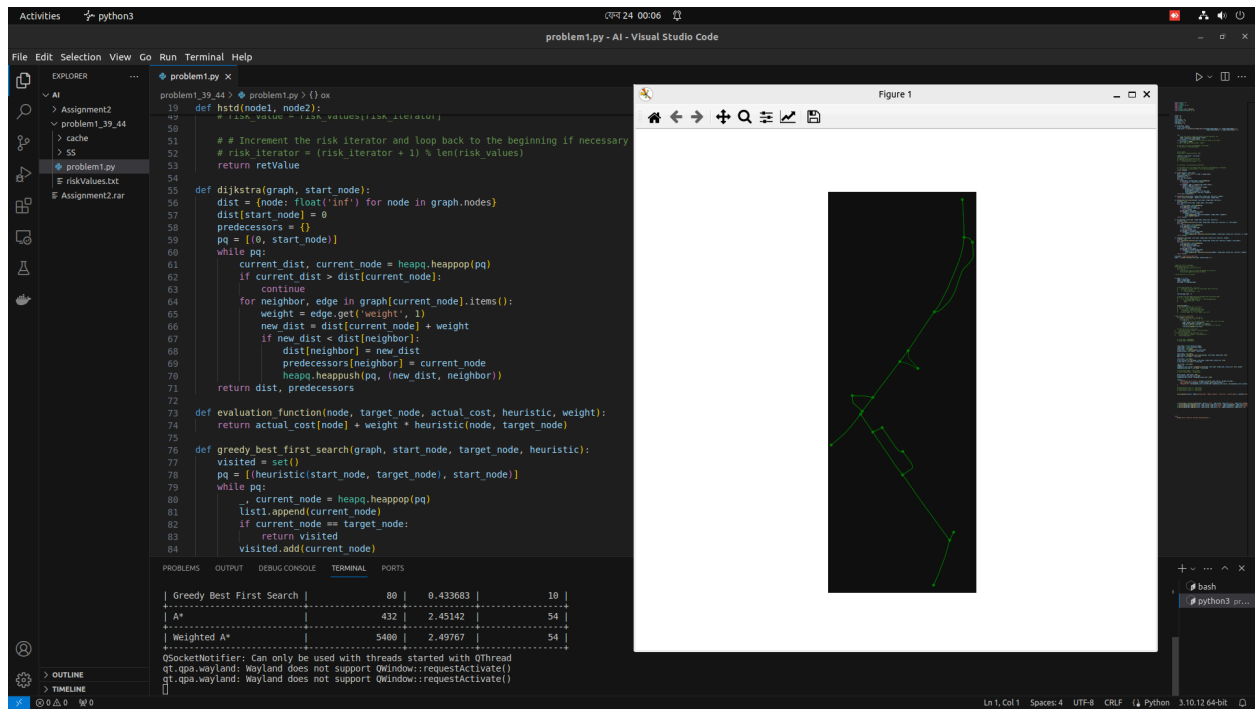


Fig:GBFS Hit Map

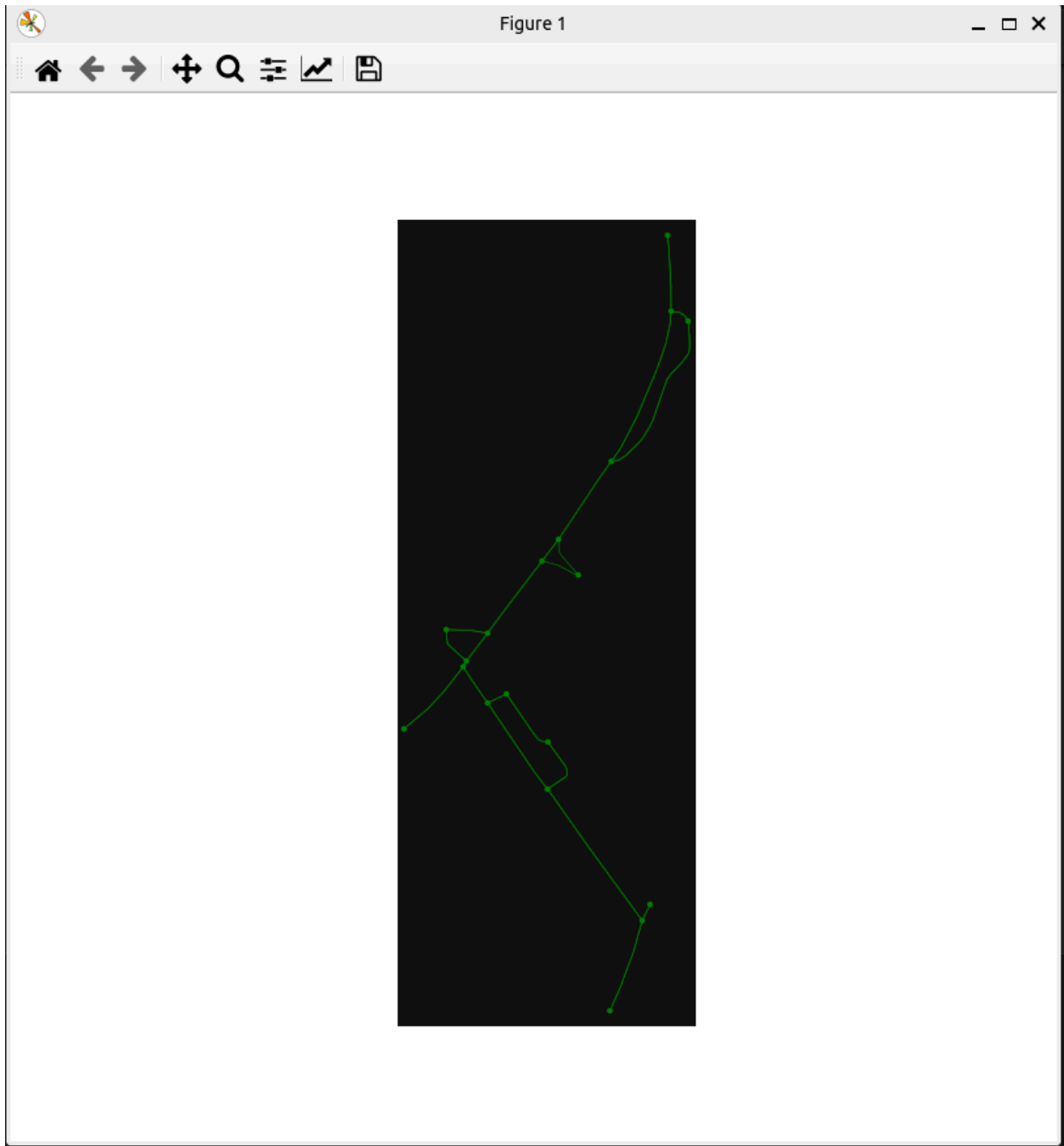


Fig:GBFS Hit Map Cropped

2.3 A* :

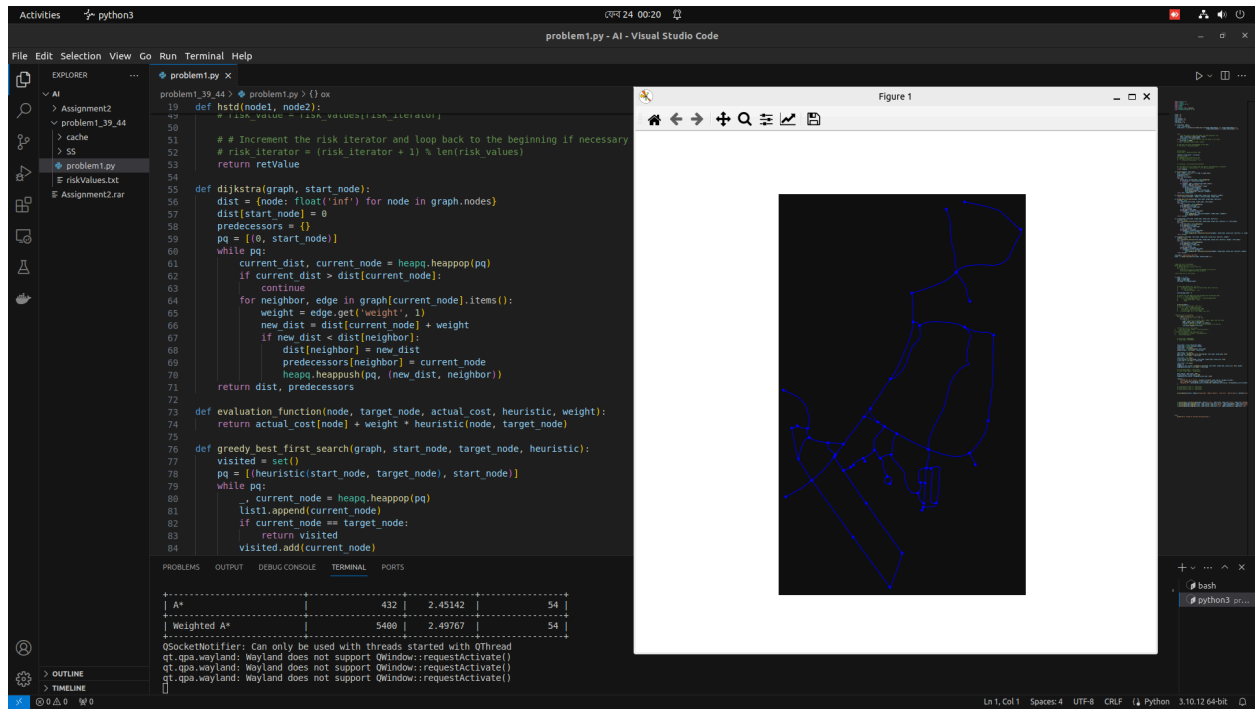


Fig:A* Hit Map

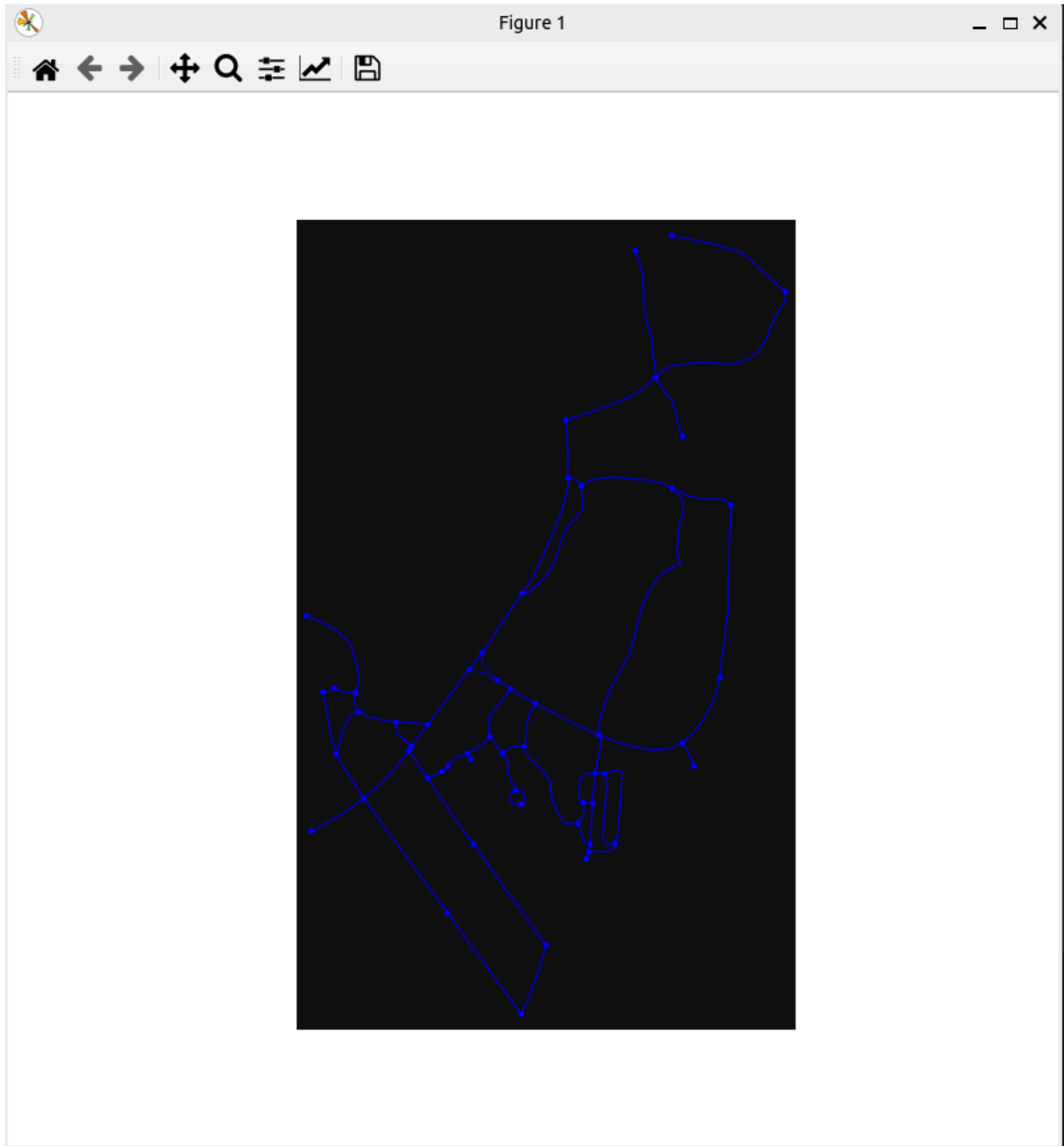


Fig:A* Hit Map Cropped

2.4 Weighted A*:

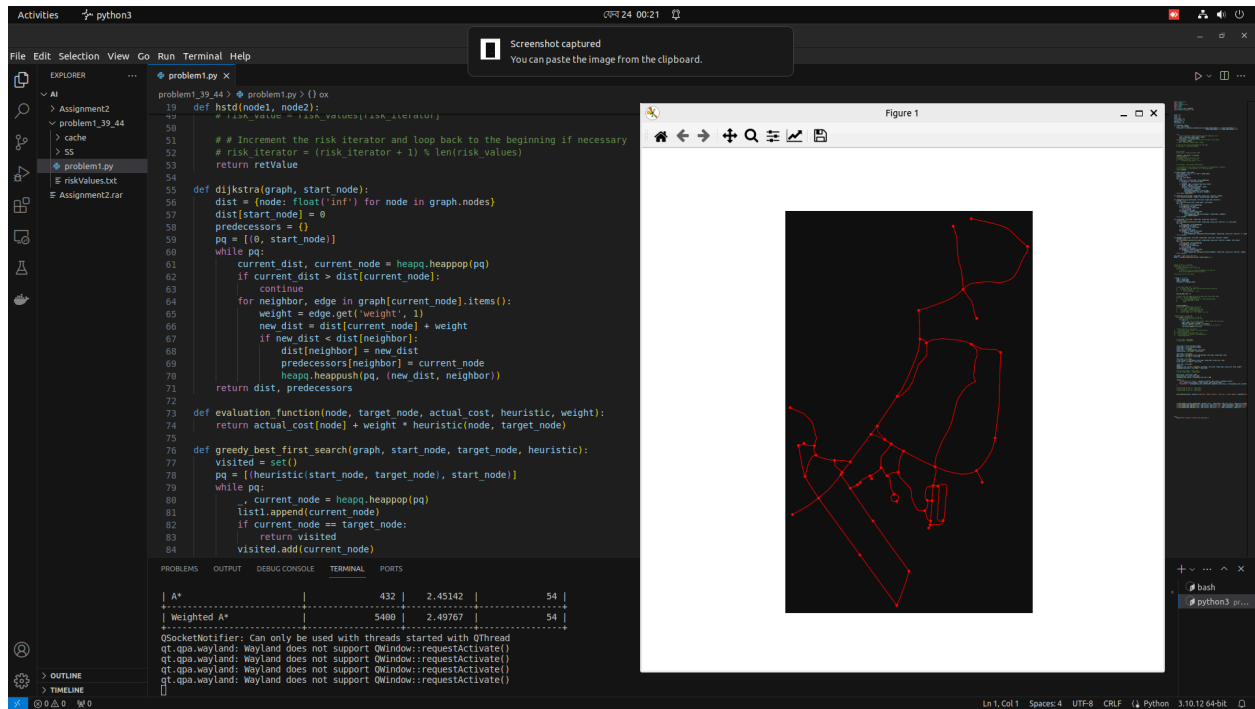
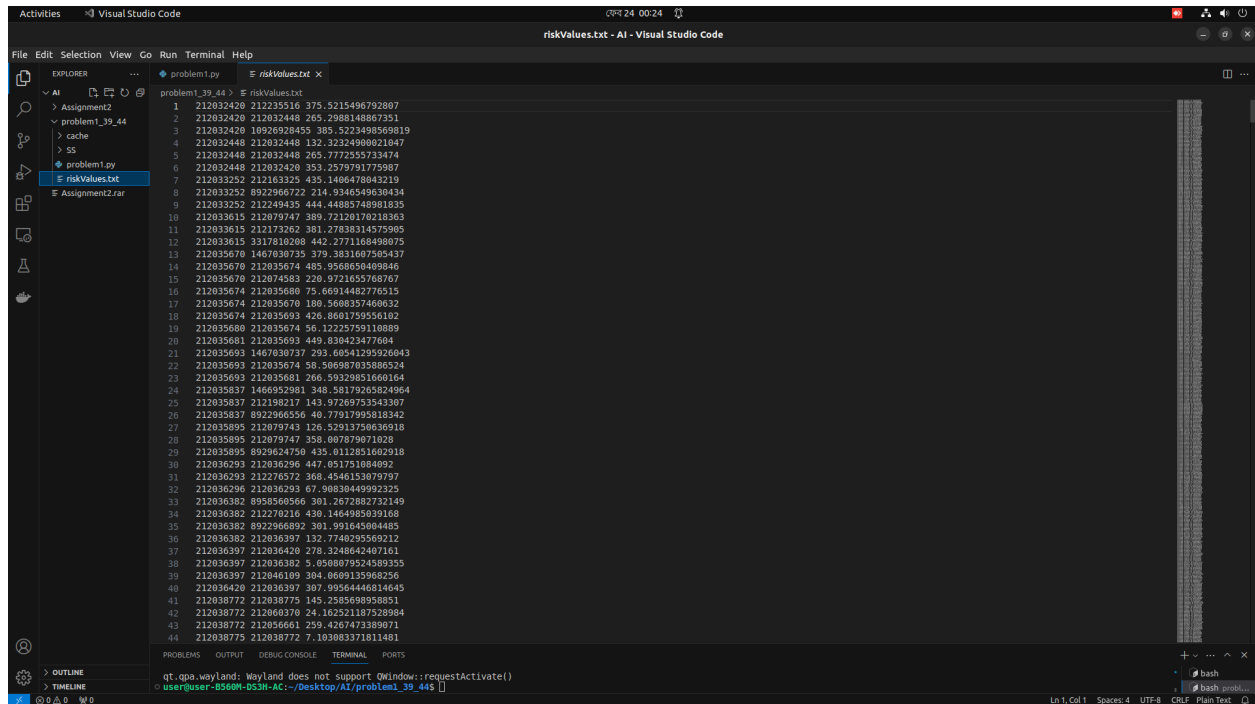


Fig:Weighted A* Hit Map



Fig:Weighted A* Hit Map Cropped

3.Risk Values:



The screenshot shows the Visual Studio Code editor with the file `riskValues.txt` open. The file contains 44 lines of numerical data, each line representing a risk value. The data is organized into columns, with the first column containing a sequence of numbers from 1 to 44. The second column contains a sequence of numbers from 212032420 to 212038775. The third column contains a sequence of numbers from 212235516 to 212066370. The fourth column contains a sequence of numbers from 375 to 24. The fifth column contains a sequence of numbers from 5215496792807 to 103003371011401. The sixth column contains a sequence of numbers from 1486478843219 to 103003371011401. The seventh column contains a sequence of numbers from 1486478843219 to 103003371011401. The eighth column contains a sequence of numbers from 1486478843219 to 103003371011401. The ninth column contains a sequence of numbers from 1486478843219 to 103003371011401. The tenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The eleventh column contains a sequence of numbers from 1486478843219 to 103003371011401. The twelfth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirteenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The fourteenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The fifteenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The sixteenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The seventeenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The eighteenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The nineteenth column contains a sequence of numbers from 1486478843219 to 103003371011401. The twentieth column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-first column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-second column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-third column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-fourth column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-fifth column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-sixth column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-seventh column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-eighth column contains a sequence of numbers from 1486478843219 to 103003371011401. The twenty-ninth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirtieth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-first column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-second column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-third column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-fourth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-fifth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-sixth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-seventh column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-eighth column contains a sequence of numbers from 1486478843219 to 103003371011401. The thirty-ninth column contains a sequence of numbers from 1486478843219 to 103003371011401. The fortieth column contains a sequence of numbers from 1486478843219 to 103003371011401. The forty-first column contains a sequence of numbers from 1486478843219 to 103003371011401. The forty-second column contains a sequence of numbers from 1486478843219 to 103003371011401. The forty-third column contains a sequence of numbers from 1486478843219 to 103003371011401. The forty-fourth column contains a sequence of numbers from 1486478843219 to 103003371011401.

Fig:Risk Values input file