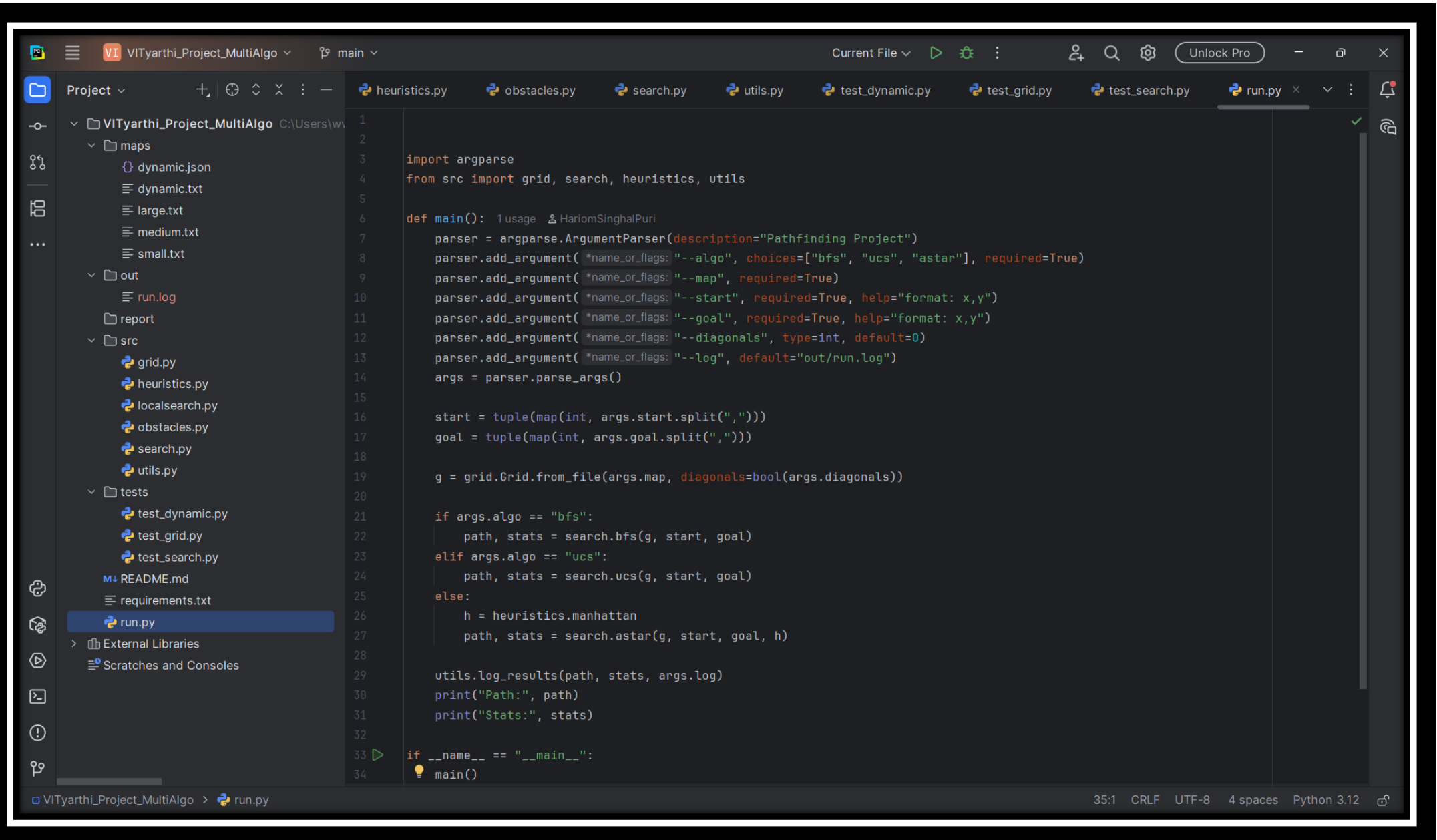


Project Preface



Test 1: BFS on small map

The screenshot displays the PyCharm IDE interface for a project named "VITyarthi_Project_MultiAlgo". The file explorer on the left shows the project structure, including a "maps" directory with "dynamic.json", "dynamic.txt", "large.txt", "medium.txt", and "small.txt", and a "src" directory with "grid.py", "heuristics.py", "localsearch.py", "obstacles.py", and "search.py". The main editor shows the "run.py" file, which contains a Python script for a pathfinding algorithm. The script defines a "main" function that takes command-line arguments for the algorithm type, map file, start point, and goal point. It uses a BFS algorithm to find the path from the start to the goal. The terminal at the bottom shows the command "python run.py --algo bfs --map maps/small.txt --start 0,0 --goal 4,4" and its output, which includes the path and statistics.

```
def main():
    usage = "usage: %s [-h] --algo ALGO --map MAP --start START --goal GOAL"
    parser = argparse.ArgumentParser(usage=usage)
    parser.add_argument("--algo", type=str, required=True)
    parser.add_argument("--map", type=str, required=True)
    parser.add_argument("--start", type=str, required=True)
    parser.add_argument("--goal", type=str, required=True)
    args = parser.parse_args()

    if args.algo == "bfs":
        path, stats = search.bfs(g, start, goal)
    elif args.algo == "ucs":
        path, stats = search.ucs(g, start, goal)
    else:
        h = heuristics.manhattan
        path, stats = search.astar(g, start, goal, h)

    utils.log_results(path, stats, args.log)
    print("Path:", path)
    print("Stats:", stats)

if __name__ == "__main__":
    main()
```

Terminal Output:

```
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo bfs --map maps/small.txt --start 0,0 --goal 4,4
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4)]
Stats: {'nodes_expanded': 20, 'path_cost': 8}
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo>
```

Test 2: UCS on medium map

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, search, and settings. The left sidebar displays the project structure, including files like `grid.py`, `heuristics.py`, `localsearch.py`, `obstacles.py`, `search.py`, `utils.py`, and a `tests` directory with `test_dynamic.py`, `test_grid.py`, and `test_search.py`. The main editor window shows the `run.py` file with the following code:

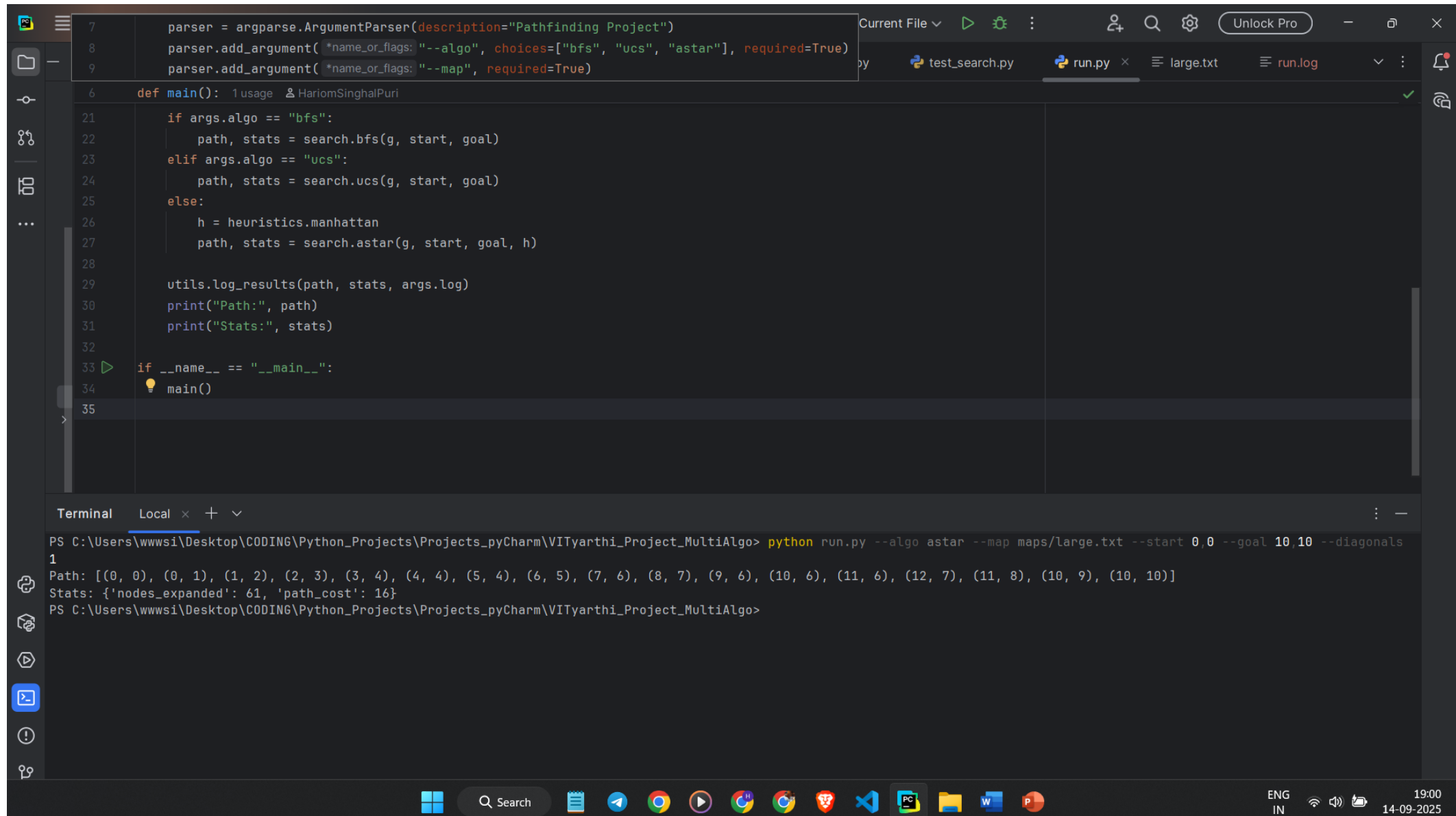
```
6 def main(): 1 usage & HariomSinghalPuri
21     if args.algo == "bfs":
22         path, stats = search.bfs(g, start, goal)
23     elif args.algo == "ucs":
24         path, stats = search.ucs(g, start, goal)
25     else:
26         h = heuristics.manhattan
27         path, stats = search.astar(g, start, goal, h)
28
29     utils.log_results(path, stats, args.log)
30     print("Path:", path)
31     print("Stats:", stats)
32
33 if __name__ == "__main__":
34     main()
35
```

The bottom terminal window shows the execution of the program. The first command runs the program with the BFS algorithm on a small map, resulting in a path and stats. The second command runs the program with the UCS algorithm on a medium map, resulting in an empty path and infinite cost.

```
PS C:\Users\wwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo bfs --map maps/small.txt --start 0,0 --goal 4,4
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4)]
Stats: {'nodes_expanded': 20, 'path_cost': 8}
PS C:\Users\wwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo ucs --map maps/medium.txt --start 2,2 --goal 7,7
Path: []
Stats: {'nodes_expanded': 70, 'path_cost': inf}
PS C:\Users\wwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo>
```

The status bar at the bottom indicates the current file is `run.py` in the `VITyarthi_Project_MultiAlgo` project, with settings for 35:1 CRLF UTF-8 4 spaces Python 3.12.

Test 3: A* with large Map allowed



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script for a pathfinding project. The script uses the argparse module to handle command-line arguments for the algorithm (bfs, ucs, astar) and a map file. The main function implements a search algorithm that selects between bfs, ucs, and astar based on the provided arguments. It uses the heapq module for priority queue operations and the heapq module for priority queue operations. The search function returns a path and statistics. The main function prints the path and statistics. The script is executed from the command line using the command: `python run.py --algo astar --map maps/large.txt --start 0,0 --goal 10,10 --diagonals 1`. The output shows the path and statistics.

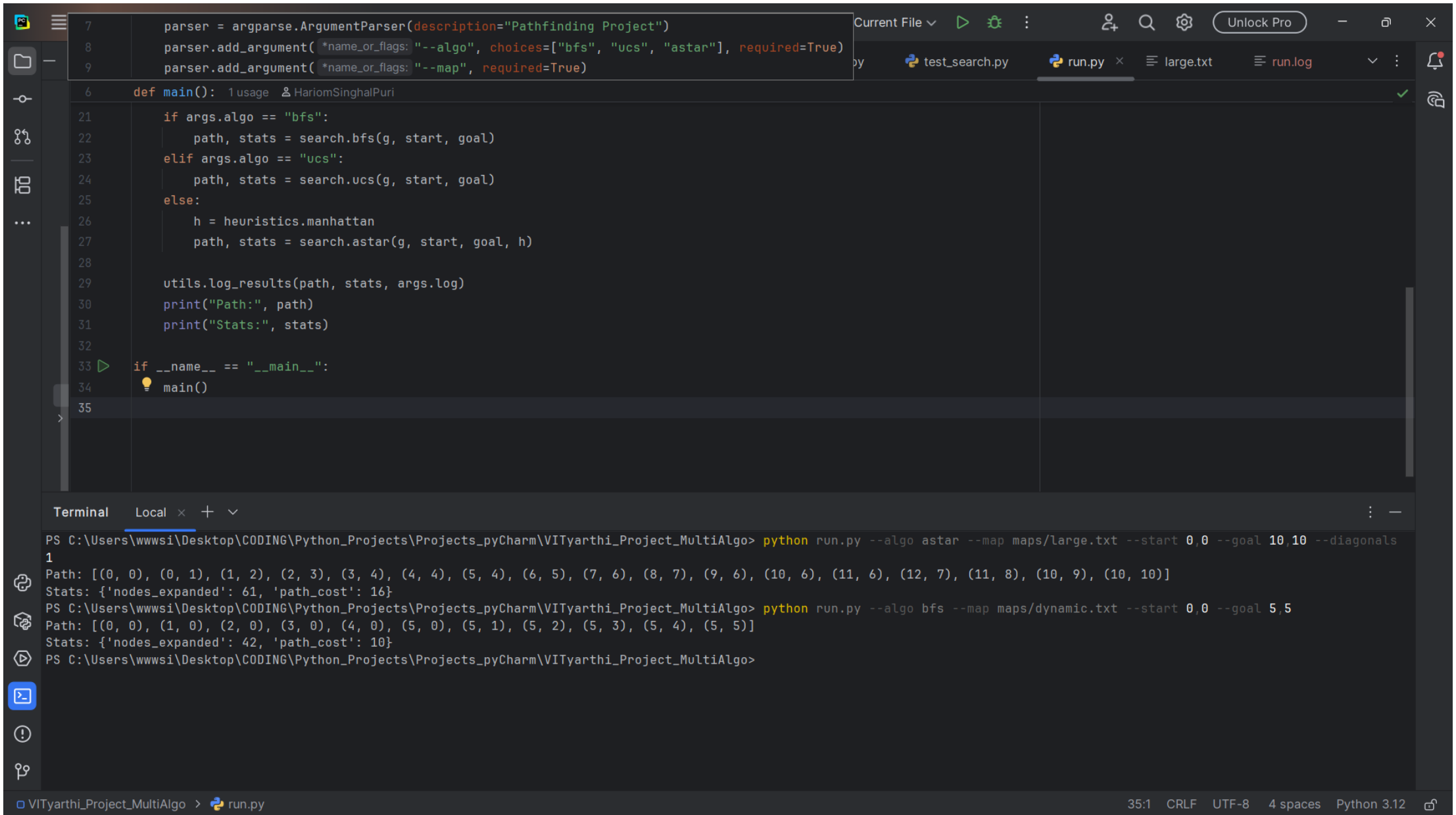
```
7 parser = argparse.ArgumentParser(description="Pathfinding Project")
8 parser.add_argument(*name_or_flags: "--algo", choices=["bfs", "ucs", "astar"], required=True)
9 parser.add_argument(*name_or_flags: "--map", required=True)

6 def main(): 1 usage: & HariomSinghalPuri
21     if args.algo == "bfs":
22         path, stats = search.bfs(g, start, goal)
23     elif args.algo == "ucs":
24         path, stats = search.ucs(g, start, goal)
25     else:
26         h = heuristics.manhattan
27         path, stats = search.astar(g, start, goal, h)
28
29     utils.log_results(path, stats, args.log)
30     print("Path:", path)
31     print("Stats:", stats)
32
33 if __name__ == "__main__":
34     main()
35
```

Terminal Local x + v

```
PS C:\Users\wwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo astar --map maps/large.txt --start 0,0 --goal 10,10 --diagonals 1
Path: [(0, 0), (0, 1), (1, 2), (2, 3), (3, 4), (4, 4), (5, 4), (6, 5), (7, 6), (8, 7), (9, 6), (10, 6), (11, 6), (12, 7), (11, 8), (10, 9), (10, 10)]
Stats: {'nodes_expanded': 61, 'path_cost': 16}
PS C:\Users\wwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo>
```

Test 4: BFS on large map with logging



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `run.py` with the following code:

```
7 parser = argparse.ArgumentParser(description="Pathfinding Project")
8 parser.add_argument(*name_or_flags: "--algo", choices=["bfs", "ucs", "astar"], required=True)
9 parser.add_argument(*name_or_flags: "--map", required=True)

6 def main():
21     if args.algo == "bfs":
22         path, stats = search.bfs(g, start, goal)
23     elif args.algo == "ucs":
24         path, stats = search.ucs(g, start, goal)
25     else:
26         h = heuristics.manhattan
27         path, stats = search.astar(g, start, goal, h)
28
29     utils.log_results(path, stats, args.log)
30     print("Path:", path)
31     print("Stats:", stats)
32
33 if __name__ == "__main__":
34     main()
35
```

The terminal window at the bottom shows the execution of the script. The first command runs the script with the A* algorithm on a large map, and the second command runs it with the BFS algorithm on a dynamic map.

```
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VItyarthi_Project_MultiAlgo> python run.py --algo astar --map maps/large.txt --start 0,0 --goal 10,10 --diagonals 1
Path: [(0, 0), (0, 1), (1, 2), (2, 3), (3, 4), (4, 4), (5, 4), (6, 5), (7, 6), (8, 7), (9, 6), (10, 6), (11, 6), (12, 7), (11, 8), (10, 9), (10, 10)]
Stats: {'nodes_expanded': 61, 'path_cost': 16}
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VItyarthi_Project_MultiAlgo> python run.py --algo bfs --map maps/dynamic.txt --start 0,0 --goal 5,5
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)]
Stats: {'nodes_expanded': 42, 'path_cost': 10}
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VItyarthi_Project_MultiAlgo>
```

The status bar at the bottom indicates the current file is `run.py` in the `VItyarthi_Project_MultiAlgo` directory, using Python 3.12 with 4 spaces indentation and UTF-8 encoding.

Test 5: A* on large map, log to custom file

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, search, and settings, along with a 'Unlock Pro' button. The editor window displays a Python script with the following code:

```
6 def main(): 1 usage 2 HariomSinghalPuri
21     if args.algo == "bfs":
22         path, stats = search.bfs(g, start, goal)
23     elif args.algo == "ucs":
24         path, stats = search.ucs(g, start, goal)
25     else:
26         h = heuristics.manhattan
27         path, stats = search.astar(g, start, goal, h)
28
29     utils.log_results(path, stats, args.log)
30     print("Path:", path)
31     print("Stats:", stats)
32
33 if __name__ == "__main__":
34     main()
35
```

The bottom terminal window shows the execution of the script with the following commands and output:

```
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo astar --map maps/large.txt --start 0,0 --goal 10,10 --diagonals 1
Path: [(0, 0), (0, 1), (1, 2), (2, 3), (3, 4), (4, 4), (5, 4), (6, 5), (7, 6), (8, 7), (9, 6), (10, 6), (11, 6), (12, 7), (11, 8), (10, 9), (10, 10)]
Stats: {'nodes_expanded': 61, 'path_cost': 16}
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo bfs --map maps/dynamic.txt --start 0,0 --goal 5,5
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)]
Stats: {'nodes_expanded': 42, 'path_cost': 10}
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo> python run.py --algo astar --map maps/large.txt --start 7,7 --goal 14,0 --log out/run.log
Path: [(7, 7), (7, 6), (8, 6), (9, 6), (10, 6), (10, 5), (10, 4), (10, 3), (10, 2), (10, 1), (10, 0), (11, 0), (12, 0), (13, 0), (14, 0)]
Stats: {'nodes_expanded': 27, 'path_cost': 14}
PS C:\Users\wwwsi\Desktop\CODING\Python_Projects\Projects_pyCharm\VITyarthi_Project_MultiAlgo>
```

The status bar at the bottom indicates the current file is 'run.py' and provides details about the file encoding (UTF-8), line endings (CRLF), and the Python version (3.12).