

Screenshots

NAME: ADHIKARI VISWA TEJA

REG N.O: 24MIM10138

TEST 1 : BFS

- BFS on small map

The screenshot shows the Visual Studio Code editor with a project named 'AIML - Project 1'. The Explorer panel on the left shows a file structure with 'maps' containing 'dynamic.json', 'dynamic.txt', 'large.txt', 'medium.txt', and 'small.txt'. The 'run.py' file is open in the editor, showing a BFS implementation. The code defines a main function that takes arguments for algorithm, map, start, and goal. It uses a grid from a file and performs a BFS search. The output in the terminal shows the path and stats for a small map.

```
run.py > ...
6 def main():
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)
```

Terminal Output:

```
PS C:\Users\viswa\Videos\AIML - Project 1> 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\viswa\Videos\AIML - Project 1>
```

- BFS on medium map

The screenshot shows the Visual Studio Code editor with the same project. The 'run.py' file is open, showing the same BFS implementation. The terminal output shows the path and stats for a medium map.

```
run.py > ...
6 def main():
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)
```

Terminal Output:

```
PS C:\Users\viswa\Videos\AIML - Project 1> python run.py --algo bfs --map maps/medium.txt --start 0,0 --goal 8,8
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (5, 1), (5, 2), (5, 3), (6, 3), (7, 3), (7, 4), (7, 5), (7, 6), (8, 6), (8, 7), (8, 8)]
Stats: {'nodes_expanded': 66, 'path_cost': 16}
PS C:\Users\viswa\Videos\AIML - Project 1>
```

- BFS on large map

```
def main():
    start = tuple(map(int, args.start.split(",")))
    goal = tuple(map(int, args.goal.split(",")))

    g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))

    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)
```

```
PS C:\Users\viswa\Videos\AIML - Project 1> python run.py --algo bfs --map maps/large.txt --start 0,0 --goal 10,10
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0), (10, 0), (11, 0), (12, 0), (12, 1), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (11, 8), (10, 8), (10, 9), (10, 10)]
Stats: {'nodes_expanded': 139, 'path_cost': 24}
PS C:\Users\viswa\Videos\AIML - Project 1>
```

TEST 2: UCS

- UCS on small map

```
def main():
    start = tuple(map(int, args.start.split(",")))
    goal = tuple(map(int, args.goal.split(",")))

    g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))

    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)
```

```
PS C:\Users\viswa\Videos\AIML - Project 1> python run.py --algo ucs --map maps/small.txt --start 0,0 --goal 4,4
Path: [(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 4), (3, 4), (4, 4)]
Stats: {'nodes_expanded': 20, 'path_cost': 8}
PS C:\Users\viswa\Videos\AIML - Project 1>
```

- UCS on medium map

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the project structure. The main editor shows the code for `run.py`, which implements a search algorithm. The terminal at the bottom shows the command `python run.py --algo ucs --map maps/medium.txt --start 0,0 --goal 8,8` and its output:

```
Path: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (5, 1), (5, 2), (5, 3), (6, 3), (7, 3), (7, 4), (7, 5), (7, 6), (8, 6), (8, 7), (8, 8)]
Stats: {'nodes_expanded': 66, 'path_cost': 16}
```

- UCS on large map

The screenshot shows the Visual Studio Code interface with the file explorer on the left. The main editor shows the code for `run.py`. The terminal at the bottom shows the command `python run.py --algo ucs --map maps/large.txt --start 0,0 --goal 10,10` and its output:

```
Path: [(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (1, 6), (2, 6), (3, 6), (4, 6), (5, 6), (6, 6), (7, 6), (8, 6), (9, 6), (10, 6), (11, 6), (12, 6), (12, 7), (12, 8), (11, 8), (10, 8), (10, 9), (10, 10)]
Stats: {'nodes_expanded': 138, 'path_cost': 24}
```

TEST 3: A* with diagonals

- A* on small map

```
run.py> main
6 def main():
15
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)
```

```
PS C:\Users\viswa\Videos\AIML - Project 1> python run.py --algo astar --map maps/small.txt --start 0,0 --goal 4,4 --diagonals 1
Path: [(0, 0), (0, 1), (1, 2), (2, 3), (3, 4), (4, 4)]
Stats: {'nodes_expanded': 6, 'path_cost': 5}
```

• A* on medium map

```
run.py> main
6 def main():
15
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)
```

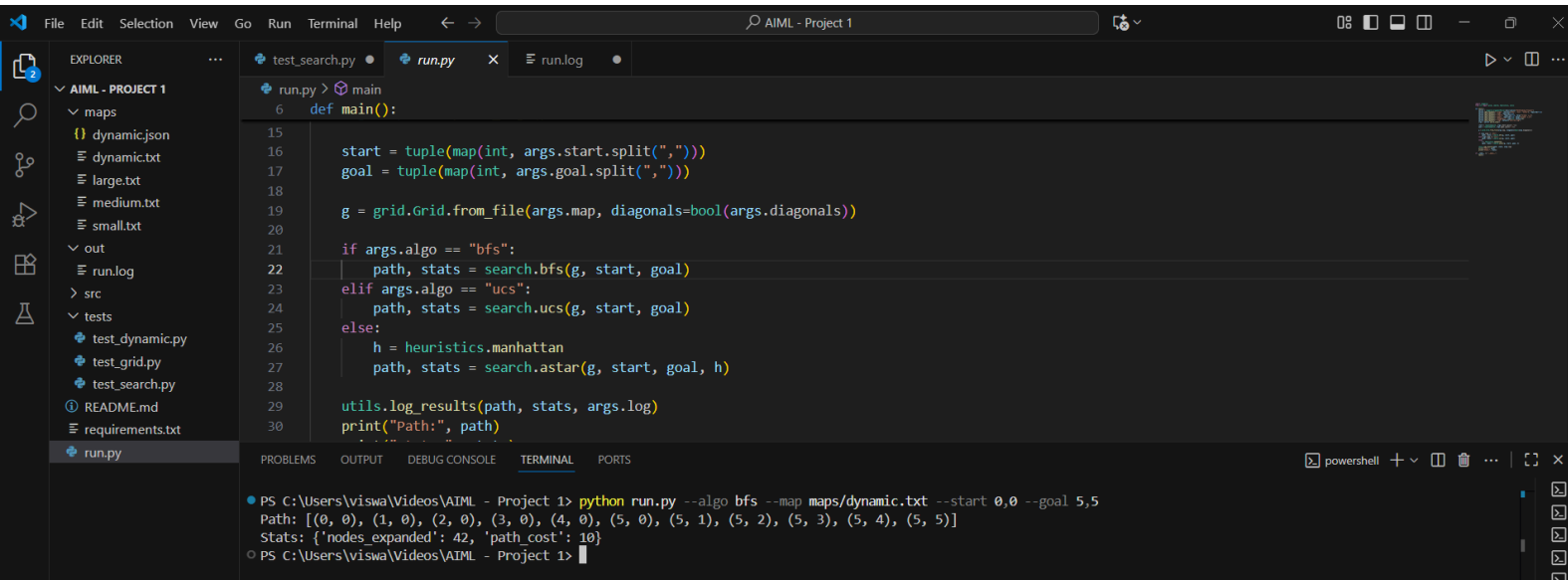
```
PS C:\Users\viswa\Videos\AIML - Project 1> python run.py --algo astar --map maps/medium.txt --start 0,0 --goal 8,8 --diagonals 1
Path: [(0, 0), (0, 1), (1, 2), (2, 2), (3, 3), (3, 4), (2, 5), (3, 6), (4, 7), (5, 8), (6, 8), (7, 9), (8, 8)]
Stats: {'nodes_expanded': 16, 'path_cost': 12}
```

• A* on large map

```
run.py> main
6 def main():
15
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)
```

```
PS C:\Users\viswa\Videos\AIML - Project 1> 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}
```

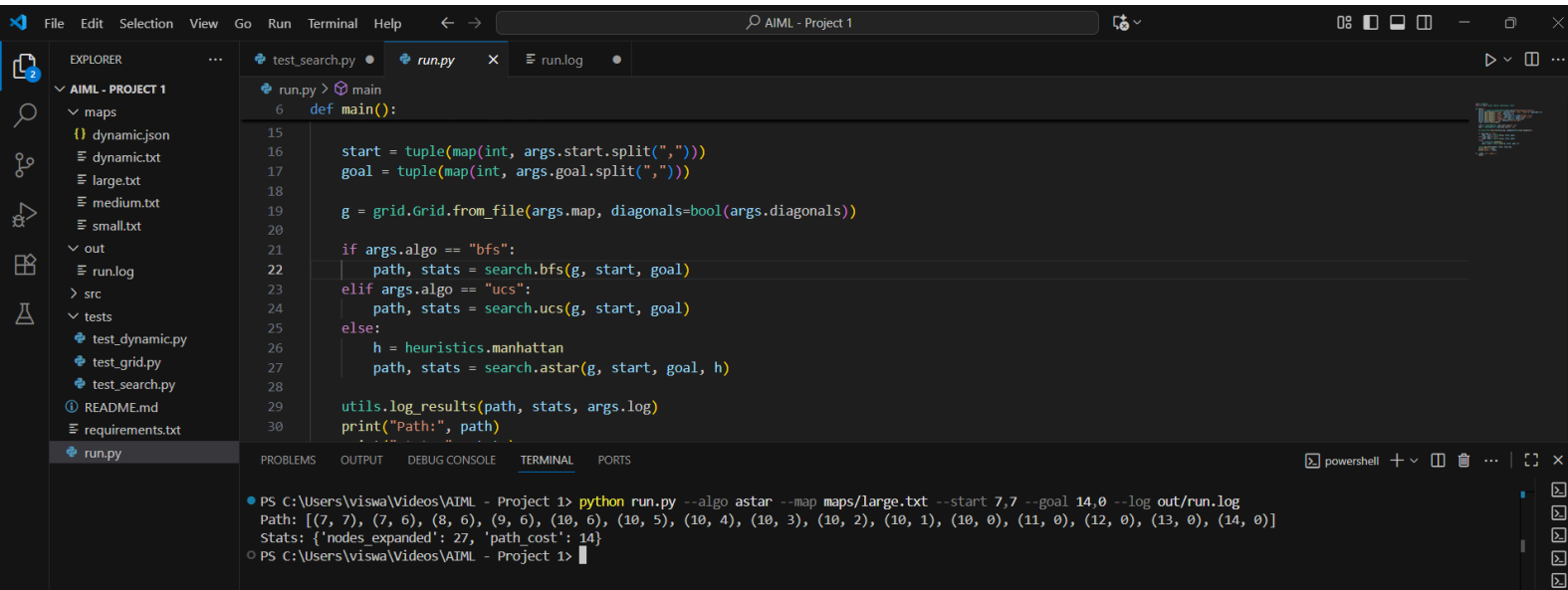
TEST 4: BFS on large map with logging



```
run.py > main
6 def main():
15
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\viswa\Videos\AIML - Project 1> 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\viswa\Videos\AIML - Project 1>
```

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



```
run.py > main
6 def main():
15
16     start = tuple(map(int, args.start.split(",")))
17     goal = tuple(map(int, args.goal.split(",")))
18
19     g = grid.Grid.from_file(args.map, diagonals=bool(args.diagonals))
20
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)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\viswa\Videos\AIML - Project 1> 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\viswa\Videos\AIML - Project 1>
```