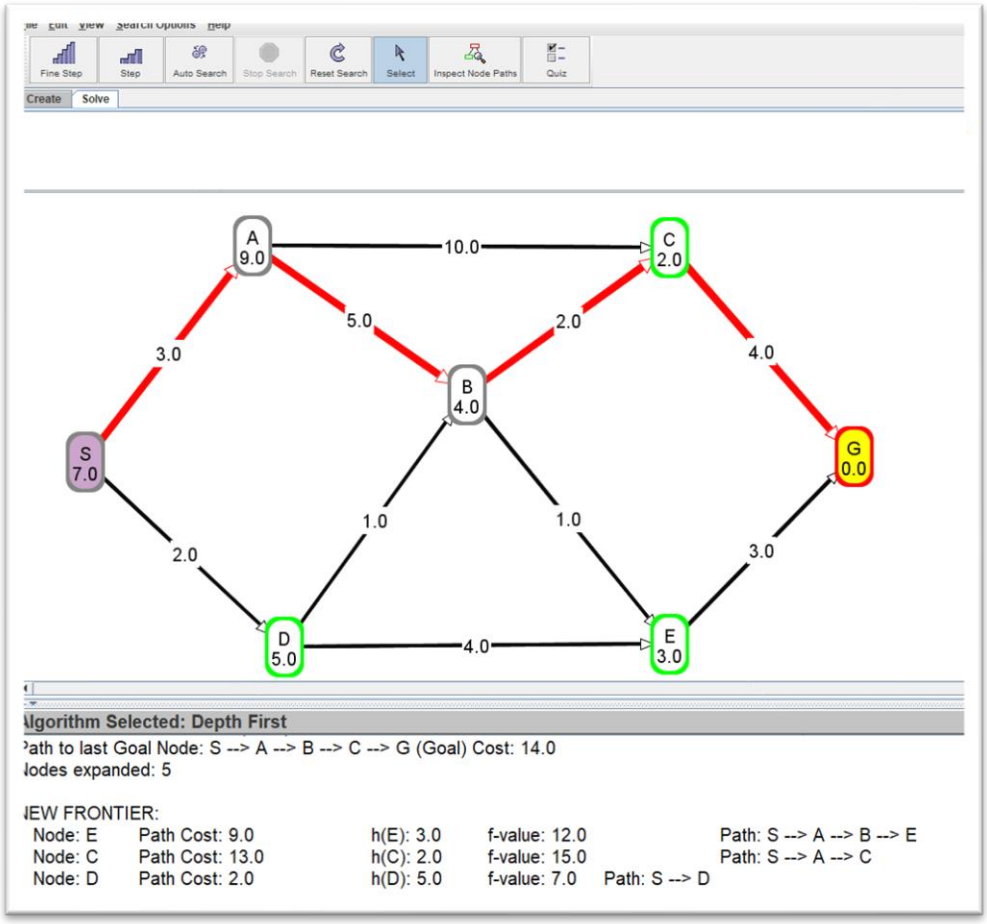
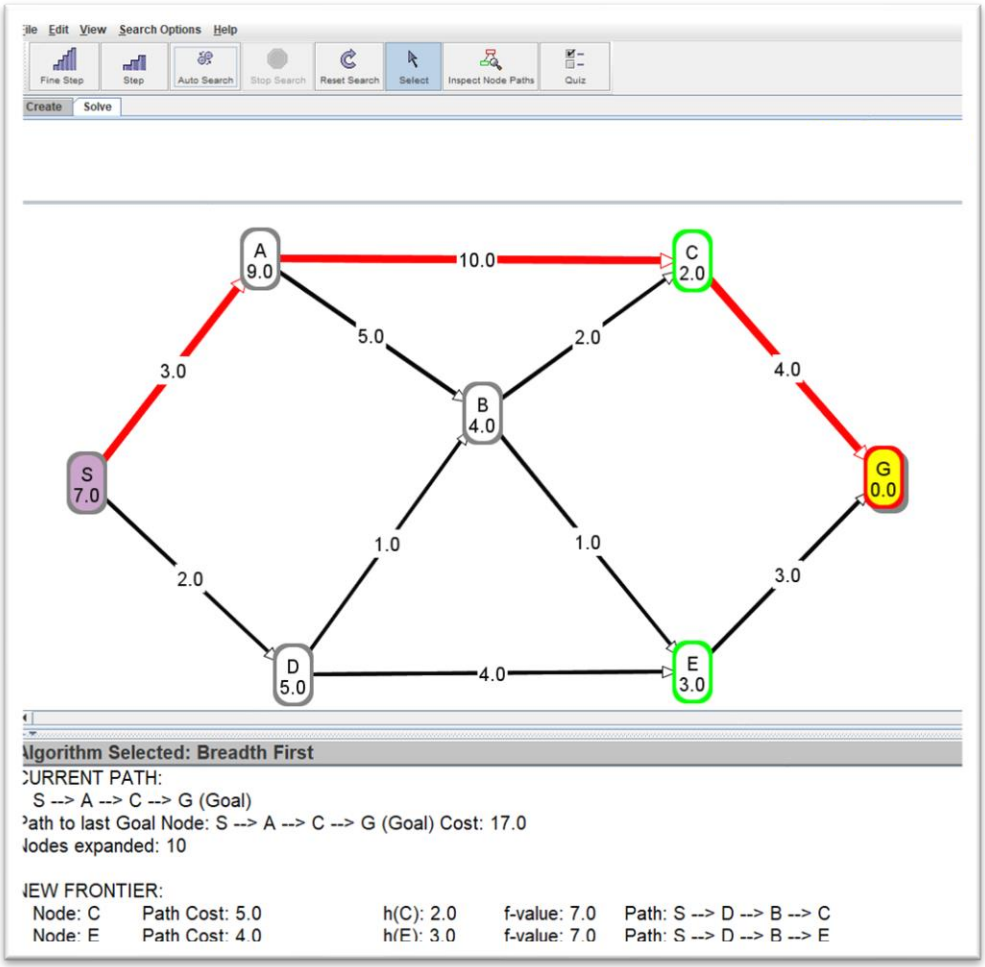
DFS


Fine Step
Step
Auto Search
Stop Search
Reset Search
Select
Inspect Node Paths
Quiz
Create
Solve
A
9.0
C
2.0
B
4.0
S
7.0
G
0.0
D
5.0
E
3.0
10.0
5.0
2.0
3.0
4.0
1.0
1.0
2.0
3.0
4.0
lgorithm Selected: Depth First
ath to last Goal Node: S --> A --> B --> C --> G (Goal) Cost: 14.0
lodes expanded: 5
EW FRONTIER:
Node: E Path Cost: 9.0 h(E): 3.0 f-value: 12.0 Path: S --> A --> B --> E
Node: C Path Cost: 13.0 h(C): 2.0 f-value: 15.0 Path: S --> A --> C
Node: D Path Cost: 2.0 h(D): 5.0 f-value: 7.0 Path: S --> D

BFS


ile Edit View Search Options Help
Fine Step
Step
Auto Search
Stop Search
Reset Search
Select
Inspect Node Paths
Quiz
Create
Solve
A
9.0
C
2.0
B
4.0
S
7.0
G
0.0
D
5.0
E
3.0
10.0
5.0
2.0
3.0
4.0
1.0
1.0
2.0
3.0
4.0
lgorithm Selected: Breadth First
URRENT PATH:
S --> A --> C --> G (Goal)
ath to last Goal Node: S --> A --> C --> G (Goal) Cost: 17.0
lodes expanded: 10
EW FRONTIER:
Node: C Path Cost: 5.0 h(C): 2.0 f-value: 7.0 Path: S --> D --> B --> C
Node: E Path Cost: 4.0 h(E): 3.0 f-value: 7.0 Path: S --> D --> B --> E

```python
# Graph definition
graph = {
    '5': ['3', '7'],
    '3': ['2', '4'],
    '7': ['8'],
    '2': [],
    '4': ['8'],
    '8': []
}

# Depth-limited search (DFS with a limit)
def dls(node, graph, goal, depth, visited):
    print(node, end=" ")
    if node == goal:
        return True
    if depth <= 0:
        return False

    visited.append(node)

    for neighbor in graph[node]:
        if neighbor not in visited:
            if dls(neighbor, graph, goal, depth - 1, visited):
                return True
    return False

# Iterative deepening search (IDS)
def ids(graph, start, goal, max_depth):
    for depth in range(max_depth):
        visited = []
        print(f"\nDepth level: {depth}")
        if dls(start, graph, goal, depth, visited):
            print("\nGoal found!")
            return
    print("\nGoal not found within depth limit.")

# Driver code
print("Following is the Iterative Deepening Search")
ids(graph, start: '5', goal: '8', max_depth: 5)  # Searching for node '8' starting from node '5'

```

```
C:\Users\TahaK\AppData\Local\Microsoft\WindowsApps\python3.11.exe C:\Users\TahaK\AppData\Roaming\JetBrains\PyCharmCE2024.1\scratches\scratch_3.py
Following is the Iterative Deepening Search

Depth level: 0
5
Depth level: 1
5 3 7
Depth level: 2
5 3 2 4 7 8
Goal found!

Process finished with exit code 0
```