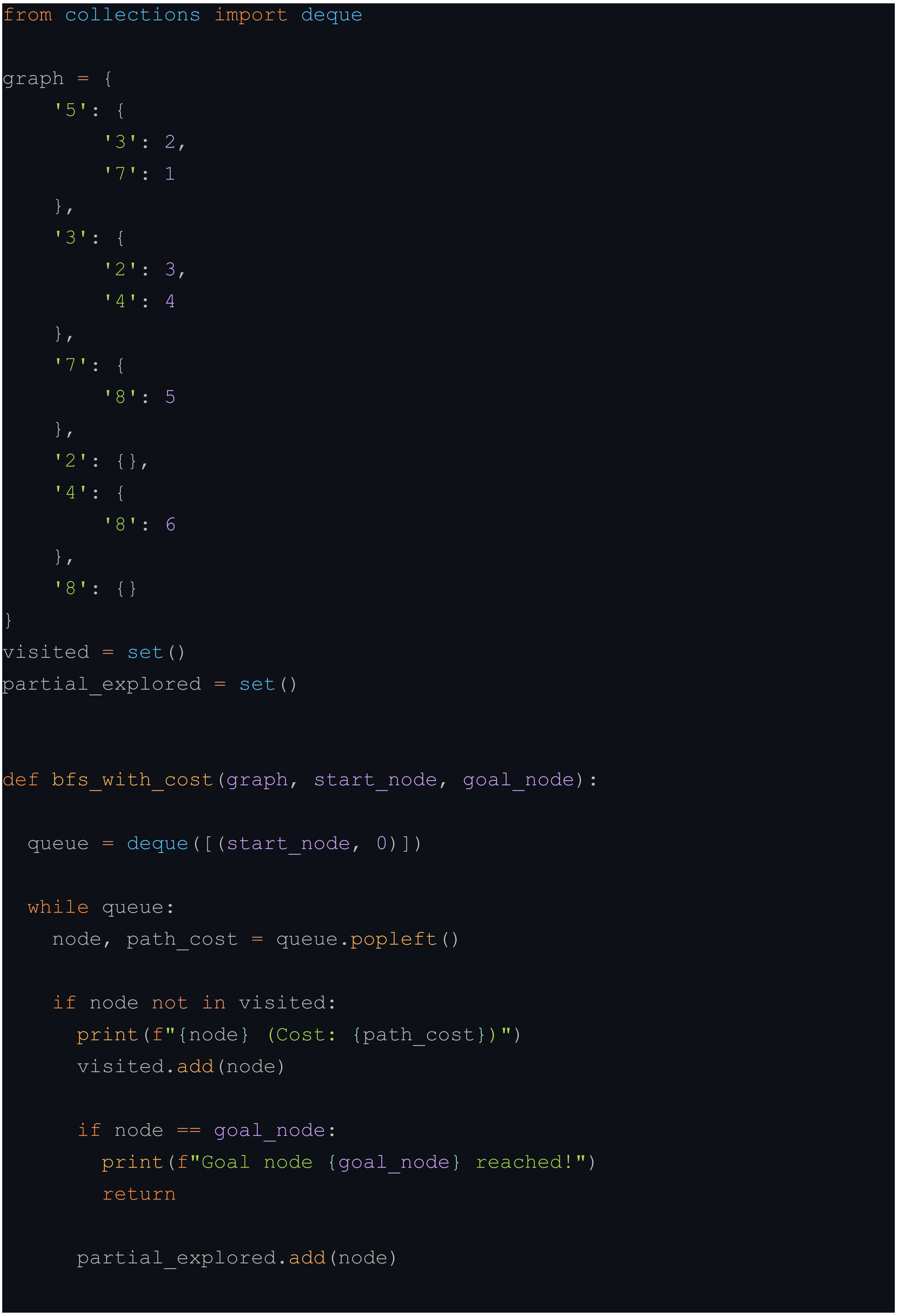
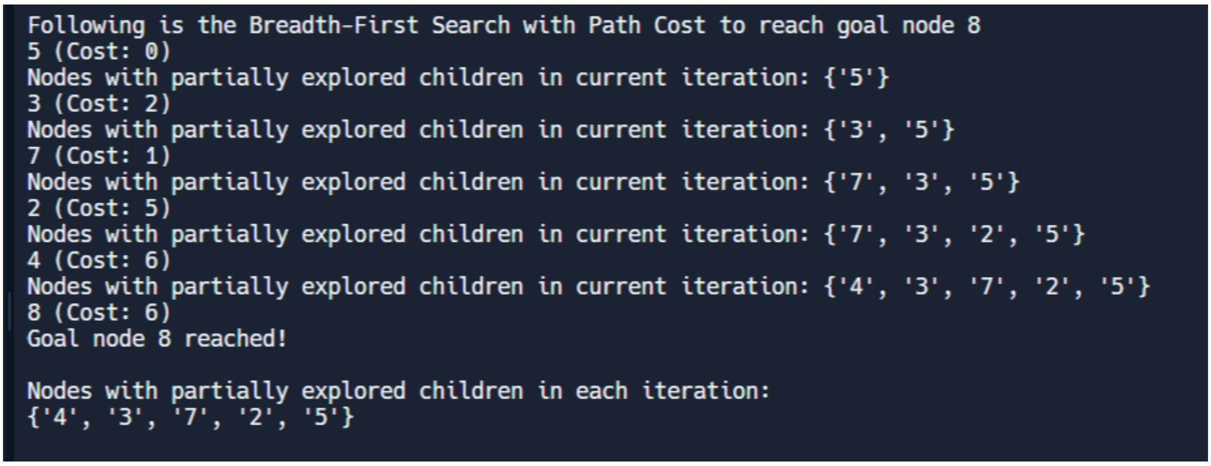
Code: (DFS)



```
from collections import deque

graph = {
    '5': {
        '3': 2,
        '7': 1
    },
    '3': {
        '2': 3,
        '4': 4
    },
    '7': {
        '8': 5
    },
    '2': {},
    '4': {
        '8': 6
    },
    '8': {}
}
visited = set()
partial_explored = set()


def bfs_with_cost(graph, start_node, goal_node):

  queue = deque([(start_node, 0)])

  while queue:
    node, path_cost = queue.popleft()

    if node not in visited:
      print(f"{node} (Cost: {path_cost})")
      visited.add(node)

      if node == goal_node:
        print(f"Goal node {goal_node} reached!")
        return

      partial_explored.add(node)
      for neighbour, edge_cost in graph[node].items():
        if neighbour not in visited:
          queue.append((neighbour, path_cost + edge_cost))

    print(
        f"Nodes with partially explored children in current iteration: {partial_explored}"
    )

goal_node = '8'
print(
    f"Following is the Breadth-First Search with Path Cost to reach goal node {goal_node}"
)
bfs_with_cost(graph, '5', goal_node)

print("\nNodes with partially explored children in each iteration:")
print(partial_explored)
```

Output:



```
Following is the Breadth-First Search with Path Cost to reach goal node 8
5 (Cost: 0)
Nodes with partially explored children in current iteration: {'5'}
3 (Cost: 2)
Nodes with partially explored children in current iteration: {'3', '5'}
7 (Cost: 1)
Nodes with partially explored children in current iteration: {'7', '3', '5'}
2 (Cost: 5)
Nodes with partially explored children in current iteration: {'7', '3', '2', '5'}
4 (Cost: 6)
Nodes with partially explored children in current iteration: {'4', '3', '7', '2', '5'}
8 (Cost: 6)
Goal node 8 reached!

Nodes with partially explored children in each iteration:
{'4', '3', '7', '2', '5'}
```