

```
2 def dfs(graph, start, target, visited, path):
3     path.append(start)
4     if start == target:
5         return path
6     visited.add(start)
7     for neighbor in graph[start]:
8         if neighbor not in visited:
9             result = dfs(graph, neighbor, target, visited,
10 path)
11             if result:
12                 return result
13     path.pop()
14     return None
15
16 def find_path(graph, start, target):
17     visited = set()
18     path = []
19     return dfs(graph, start, target, visited, path)
20
21 graph = {
22     'A': ['B', 'C'],
23     'B': ['D', 'E'],
24     'C': [],
25     'D': [],
26     'E': ['F'],
27     'F': []
28 }
29
30 result = find_path(graph, 'A', 'F')
31
32 if result:
33     print("Path from A to F:", " -> ".join(result))
34 else:
35     print("No path found.")
```

Path from A to F: A -> B -> E -> F

[Program finished]