

implement bfs with the input of graph and the goal node to be searched, your output will show the path from the root node to goal node only

```
In [1]: graph = {'A': ['B', 'C'],
                'B': ['A', 'D', 'E'],
                'C': ['F', 'G', 'A'],
                'D': ['B'],
                'E': ['H', 'B'],
                'F': ['C'],
                'G': ['C'],
                'H': ['E']}
                }
```

```
In [2]: def dfs(graph, start, end, route, list):
        route += [start]
        if start == end:
            list.extend(route)
        else:
            for node in graph[start]:
                if node not in route:
                    dfs(graph, node, end, route, list)

    def dfs_route(graph, start, end):
        list = []
        dfs(graph, start, end, [], list)
        return list

    print(dfs_route(graph, 'A', 'G'))

['A', 'B', 'D', 'E', 'H', 'C', 'F', 'G']
```

```
In [ ]:
```