

Báo cáo bài tập thực hành 1

Cài đặt thuật toán tìm kiếm DFS và BFS bằng Python

I. Báo cáo tiến độ hoàn thành

Nội dung	Đã Hoàn Thành	Chưa Hoàn Thành
1. Cài đặt lại source code mẫu BFS	Đã hoàn thành	
2. Kết quả thực thi BFS với 3 đồ thị trong đề bài	Đã hoàn thành	
3. Viết code thực thi DFS	Đã hoàn thành	
4. Thực thi code DFS với 3 đồ thị	Đã hoàn thành	

II. Kết quả chạy chương trình

1. Cài đặt lại source code mẫu BFS

```
Baitap_BFS.py > ...
1 #cài đặt lại code mẫu BFS
2 from importlib.resources import path
3
4 print("Đây là kết quả chạy 2 source code mẫu: ")
5 graph = {
6     'A':['B','C'],
7     'B':['D','E'],
8     'C':['F'],
9     'D':[],
10    'E':['F'],
11    'F':[]
12 }
13 visited = []
14 queue = []
15 def bfs_1(visited,graph,start,end):
16     visited.append(start)
17     queue.append(start)
18
19     while queue:
20         s= queue.pop()
21         print(s,end=" ")
22         if s == end:
23             return
24         for neighbour in graph[s]:
25             if neighbour not in visited:
26                 visited.append(neighbour)
27                 queue.append(neighbour)
28 bfs_1(visited,graph,'A','F')
29 print("")

Baitap_BFS.py > bfs
4/ queue.append(neighbour)
28 bfs_1(visited,graph,'A','F')
29 print("")
30 #=====BFS có xuất ra đường đi=====
31 graph = {
32     '1':['2','3','4'],
33     '2':['5','6'],
34     '5':['9','10'],
35     '4':['7','8'],
36     '7':['11','12']
37 }
38 def bfs(graph,start,end):
39     visited = []
40     queue = []
41
42     queue.append([start])
43     while queue:
44         path = queue.pop(0)
45         node = path[-1]
46         if node == end:
47             return path
48         for neighbour in graph.get(node,[]):
49             if neighbour not in visited:
50                 new_path = list(path)
51                 new_path.append(neighbour)
52                 queue.append(new_path)
53 print("BFS có xuất ra đường đi:")
54 print(bfs(graph,'1','11'))
55 print("=====")
```

2. Kết quả thực thi BFS với 3 đồ thị trong đề bài

```
PROBLEMS  OUTPUT  JUPYTER  DEBUG CONSOLE  TERMINAL

Đây là kết quả chạy 2 source code mẫu:
A C F
BFS có xuất ra đường đi:
['1', '4', '7', '11']
=====
Kết quả của đồ thị thứ 1:
Xuất ra đường đi:
['S', 'E', 'R', 'F', 'G']

=====
Kết quả của đồ thị thứ 2:
Xuất ra đường đi:
['s', 'f', 'p', 'q', 'r', 't', 'g']

=====
Kết quả của đồ thị thứ 3:
Xuất ra đường đi:
['A', 'B', 'D', 'G']

=====
PS D:\document\AI\AIBase\Lab_01\21880038>
```

3. Viết code thực thi DFS

```
Baitap_DFS.py > ...
1  #cài đặt DFS
2  from importlib.resources import path
3  from inspect import stack
4
5
6  # Set to keep track of visited nodes of graph.
7
8  def dfs(graph, root_node, end_node):
9      traversed = [root_node]
10     stack = [root_node]
11     while stack:
12         vertex = stack[-1]
13         if vertex == end_node:
14             traversed.extend(vertex)
15             return traversed
16         if vertex not in traversed:
17             traversed.extend(vertex)
18             pop = True
19             for adjacent in graph[vertex]:
20                 if adjacent not in traversed:
21                     stack.extend(adjacent)
22                     pop = False
23                     break
24             if pop:
25                 stack.pop()
26     return traversed
```

4. Thực thi code DFS với 3 đồ thị

```
> & "C:/Program Files/Python39/python.exe" d:/document/AI/AIBase/Lab_01/21880038/Baitap_DFS.py
Đây là kết quả đồ thị thứ 1-DFS:
['S', 'D', 'B', 'A', 'C', 'E', 'R', 'F', 'G']
=====
Đây là kết quả đồ thị thứ 2-DFS:
['s', 'f', 'p', 'q', 'r', 't', 'g']
=====
Đây là kết quả đồ thị thứ 3-DFS:
['A', 'B', 'D', 'E', 'G']
=====
PS D:\document\AI\AIBase\Lab_01\21880038>
```