

BÀI THỰC HÀNH – CSTTNT

BUỔI 3- GIẢI THUẬT BFS VÀ DFS

Bài 1. Viết chương trình khai báo đồ thị và duyệt đồ thị theo giải thuật Breadth-First Search (BFS) biết rằng BFS làm việc theo cơ chế Queue (hàng đợi) – FIFO (First In First Out). Sau đó xuất kết quả duyệt đồ thị ra màn hình.

```
#Khai báo đồ thị
graph = {'A':['B', 'C'],
        'B':['D', 'E'],
        'C':['E', 'I'],
        'D':['F'],
        'E':['F', 'J'],
        'F':['G', 'H'],
        'G':[], 'H':[],
        'I':['K'],
        'J':[], 'K':[]
        }

#1. Khai báo giải thuật Breadth-First Search(BFS) để duyệt đồ thị
def BFS(graph, start):
    queue=[start] #queue được khai báo dạng list, giá trị ban đầu là start
    visited=[]

    while queue: #queue khác rỗng
        node=queue.pop(0) #lấy node đầu tiên trong queue
        if node in visited:
            continue
        visited.append(node)
        for next_node in graph[node]: #triển khai các con (next_node) của node
            queue.append(next_node)
    return visited

#2. Gọi hàm BFS
print("Ket qua duyet do thi (giai thuat BFS):")
print(BFS(graph, 'A'))
```

Bài 2. Dựa và giải thuật BFS. Sinh viên viết chương trình khai báo đồ thị và duyệt đồ thị theo giải thuật Depth-First Search (DFS) biết rằng DFS làm việc theo cơ chế Stack (ngăn xếp) – LIFO (Last In First Out). Sau đó xuất kết quả duyệt đồ thị ra màn hình.

Bài 3. Viết chương trình cải tiến giải thuật Breadth-First Search (BFS) để in ra đường đi từ vị trí **start** đến vị trí **end** trong đồ thị.

```

#Khái báo đồ thị
graph = {'A':['B', 'C'],
        'B':['D', 'E'],
        'C':['E', 'I'],
        'D':['F'],
        'E':['F', 'J'],
        'F':['G', 'H'],
        'G':[], 'H':[],
        'I':['K'],
        'J':[], 'K':[]}

#1. Khai báo giải thuật Breadth-First Search(BFS) để in đường đi từ Start đến End
def BFS_Path(graph, start, end):
    queue=[(start, [start])] #queue được khai báo dạng list, giá trị ban đầu là start và cha của start

    while queue: #queue khác rỗng
        node, path=queue.pop(0) #lấy node đầu tiên trong queue
        for next_node in graph[node]: #triển khai các con (next_node) của node
            if next_node in path:
                continue
            elif next_node == end:
                return path + [next_node]
            else:
                queue.append((next_node, path + [next_node]))#queue lưu giá trị của next_node và đường dẫn đến next_node

#2. Gọi hàm BFS_Path
start = input('Nhập vị trí start = ')
end = input('Nhập vị trí end = ')
print("Đường đi từ "+start+" đến "+end+" : ")
print(BFS_Path(graph,start, end))

```

Bài 4. Dựa vào Bài 3, sinh viên viết chương trình cải tiến giải thuật Depth-First Search (DFS) để in ra đường đi từ vị trí **start** đến vị trí **end** trong đồ thị.