

Mẫu:

Thuật toán BFS (Duyệt theo chiều rộng)										
Đỉnh	1	2	3	4	5	6	7	Hàng đợi	Đỉnh	Thứ tự duyệt
1		1		1	1				1	
2	1		1	1		1			2	
3		1							3	
4	1	1			1	1			4	
5	1			1		1	1		5	
6		1		1	1				6	
7					1				7	

Đưa 1 đỉnh S bất kỳ vào Hàng Đợi

While hàng đợi chưa rỗng

```

{
    u = Lấy đỉnh ở đầu Hàng Đợi ra
    if (u đã duyệt) continue;
    Duyệt u (vd: in u ra màn hình)
    Đánh dấu u đã duyệt
    for các đỉnh kề v của u
        if v chưa được duyệt
            Đưa v vào Hàng Đợi
}

```

Vai trò của duyệt đồ thị là gì? -- Kiểm tra tính liên thông của đồ thị & Tìm kiếm trạng thái mục tiêu trong bài toán tìm kiếm không gian trạng thái (AI)

Đồ thị liên thông: giữa 2 cặp đỉnh bất kỳ trong đồ thị, luôn tìm được đường đi

Lần lặp thứ 1

Đưa 1 đỉnh s=1 bất kỳ vào Hàng Đợi

While hàng đợi chưa rỗng // true

```

{
    u=1 Lấy đỉnh ở đầu Hàng Đợi ra
    if (u=1 đã duyệt) // false
        continue;
    Duyệt 1 (vd: in 1 ra màn hình)
    Đánh dấu 1 đã duyệt
    Đỉnh kề của 1 = {2, 4, 5}
    for các đỉnh kề v của u
        v=2
        if v=2 chưa được duyệt // true
            Đưa v vào Hàng Đợi
        v=4
        if v=4 chưa được duyệt // true
            Đưa v vào Hàng Đợi
        v=5
        if v=5 chưa được duyệt // true
            Đưa v vào Hàng Đợi
}

```

Thuật toán DFS (Duyệt theo chiều sâu) cài đặt bằng Đệ Quy/Ngăn xếp										
Đỉnh	1	2	3	4	5	6	7	Ngăn xếp	Đỉnh	Thứ tự duyệt
1		1	1		1				1	1
2	1			1		1			2	2
3	1						1		3	6
4		1							4	3
5	1					1			5	5
6		1			1				6	4
7			1						7	7

void DFS Đệ Quy (Graph *G, int u)

```

{
    if (u đã duyệt) return;
    Duyệt u (vd: in u ra màn hình)
    Đánh dấu u đã duyệt
    Các đỉnh kề của u={v1, v2,...}
    for các đỉnh kề v của u
        DFS Đệ Quy (Graph *G, int v1)
}

```

Vai trò của duyệt đồ thị là gì? -- Kiểm tra tính liên thông của đồ thị & Tìm kiếm trạng thái mục tiêu trong bài toán tìm kiếm không gian trạng thái (AI)

Đồ thị liên thông: giữa 2 cặp đỉnh bất kỳ trong đồ thị, luôn tìm được đường đi

void DFS Đệ Quy (Graph *G, int 2)

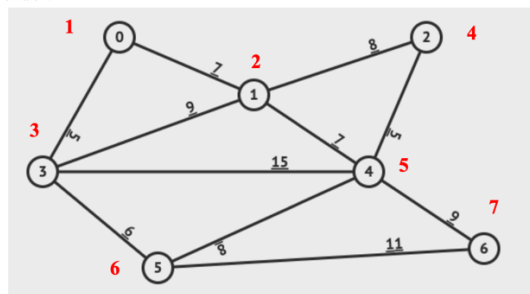
```

{
    if (2 đã duyệt) // false
        return;
    Duyệt 2 (vd: in 2 ra màn hình)
    Đánh dấu 2 đã duyệt
    Các đỉnh kề của 2 = {(1),(4),(6)}
    for các đỉnh kề v của u
        DFS Đệ Quy (Graph *G, int 1)
}

```

Mẫu đáp án:

Students visualize BFS, DFS algorithms by drawing the graph and then labeling vertices with their traversal order.



Graph traversal using BFS algorithm

Bài 1

Cho danh sách cạnh sau:

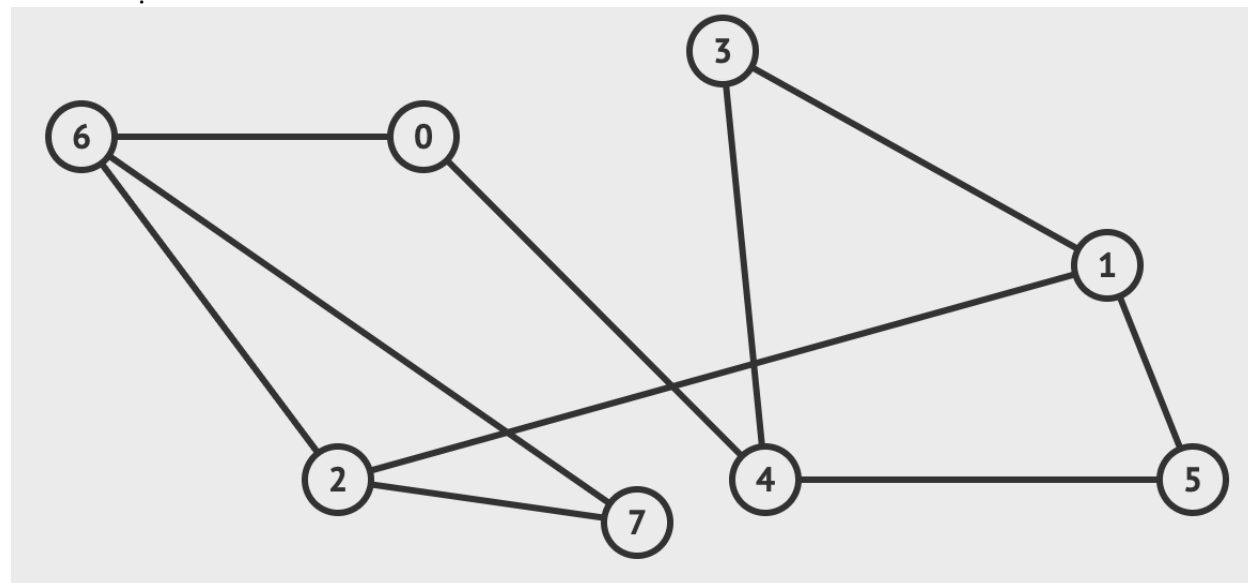
Adjacency List				
0:	1	2		
1:	0	2	3	
2:	0	1	4	6
3:	1	4	8	
4:	2	3	5	8
5:	4	6	8	
6:	2	5	7	
7:	6			
8:	3	4	5	

a/ Hãy vẽ đồ thị

b/ Duyệt các đỉnh đồ thị theo thuật toán DFS và BFS

Bài 2

Cho đồ thị sau



a/ Hãy vẽ ma trận đỉnh và danh sách kề.

b/ Duyệt các đỉnh đồ thị theo thuật toán DFS và BFS

Bài 3

Adjacency Matrix										
	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	0	1	0	1	0
2	0	0	0	1	0	0	0	1	0	0
3	0	0	1	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	1
5	1	0	0	0	0	0	1	0	0	0
6	0	1	0	0	0	1	0	0	0	0
7	1	0	1	0	0	0	0	0	0	0
8	0	1	0	1	0	0	0	0	0	1
9	0	0	0	0	1	0	0	0	1	0

a/ Hãy vẽ ra đồ thị trực quan.

b/ Duyệt các đỉnh đồ thị theo thuật toán DFS và BFS