

## MỤC LỤC

Sử dụng ngôn ngữ lập trình C++ viết thuật toán BFS và DFS để tìm đường đi giữa 2 đỉnh trên đồ thị .....	2
<b>Breadth First Search (BFS):</b> .....	2
1. Tìm kiếm trên ma trận kề .....	2
2. Tìm kiếm trên danh sách cạnh.....	3
3. Tìm kiếm trên danh sách kề.....	4
<b>Depth First Search (DFS):</b> .....	5
1. Tìm kiếm trên ma trận kề .....	5
2. Tìm kiếm trên danh sách cạnh.....	6
3. Tìm kiếm trên danh sách kề.....	7

### TÌM ĐƯỜNG ĐI GIỮA CÁC ĐỈNH TRÊN ĐỒ THỊ:

#### ❖ Kiểm nghiệm thuật toán:

Xác định đường đi từ đỉnh 1 đến đỉnh 13 trên đồ thị gồm 13 đỉnh được biểu diễn dưới dạng ma trận kề theo thuật toán **DFS**.

- Hãy cho biết kết quả thực hiện thuật toán.
- Chỉ rõ thứ tự thực hiện

0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	1	1	0	0	0	1	0
0	1	0	0	1	0	1	0	0	0	0	1	0
0	0	0	1	1	1	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	1	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0

## Sử dụng ngôn ngữ lập trình C++ viết thuật toán BFS và DFS để tìm đường đi giữa 2 đỉnh trên đồ thị

### Breadth First Search (BFS):

#### 1. Tìm kiếm trên ma trận kề

Input:

- Nhập số đỉnh  $n$  của đồ thị
- Nhập ma trận kề  $n \times n$
- Nhập đỉnh bắt đầu  $s$
- Nhập đỉnh kết thúc  $t$

Output:

- Đường đi từ đỉnh  $s$  đến đỉnh  $t$

Example:

Input	Output
13 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 13	1 2 6 12 10 13

## 2. Tìm kiếm trên danh sách cạnh

Input:

- Nhập số cạnh  $m$  của đồ thị
- Nhập danh sách  $m$  cạnh
- Nhập đỉnh bắt đầu  $s$
- Nhập đỉnh kết thúc  $t$

Output:

- Đường đi từ đỉnh  $s$  đến đỉnh  $t$

Example:

Input	Output
24 1 2 1 3 1 4 2 3 2 4 2 6 3 4 3 5 4 7 5 6 5 7 5 8 5 12 6 7 6 12 7 8 8 12 9 10 9 11 9 13 10 11 10 12 10 13 11 13 1 13	1 2 6 12 10 13

### 3. Tìm kiếm trên danh sách kề

Input:

- Nhập số đỉnh  $n$  của đồ thị
- Nhập danh sách kề của  $n$  đỉnh
- Nhập đỉnh bắt đầu  $s$
- Nhập đỉnh kết thúc  $t$

Output:

- Đường đi từ đỉnh  $s$  đến đỉnh  $t$

Example:

Input	Output
13 1: 2 3 4 2: 1 3 4 6 3: 1 2 4 5 4: 1 2 3 7 5: 3 6 7 8 12 6: 2 5 7 12 7: 4 5 6 8 8: 5 7 12 9: 10 11 13 10: 9 11 12 13 11: 9 10 13 12: 5 6 8 10 13: 9 10 11 1 13	1 2 6 12 10 13

## Depth First Search (DFS):

### 1. Tìm kiếm trên ma trận kề

Input:

- Nhập số đỉnh  $n$  của đồ thị
- Nhập ma trận kề  $n \times n$
- Nhập đỉnh bắt đầu  $s$
- Nhập đỉnh kết thúc  $t$

Output:

- Đường đi từ đỉnh  $s$  đến đỉnh  $t$

Example:

Input	Output
13 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 13	1 2 3 4 7 5 6 12 10 9 11 13

## 2. Tìm kiếm trên danh sách cạnh

Input:

- Nhập số cạnh  $m$  của đồ thị
- Nhập danh sách  $m$  cạnh
- Nhập đỉnh bắt đầu  $s$
- Nhập đỉnh kết thúc  $t$

Output:

- Đường đi từ đỉnh  $s$  đến đỉnh  $t$

Example:

Input	Output
24	1 2 3 4 7 5 6 12 10 9 11 13
1 2	
1 3	
1 4	
2 3	
2 4	
2 6	
3 4	
3 5	
4 7	
5 6	
5 7	
5 8	
5 12	
6 7	
6 12	
7 8	
8 12	
9 10	
9 11	
9 13	
10 11	
10 12	
10 13	
11 13	
1	
13	

### 3. Tìm kiếm trên danh sách kề

Input:

- Nhập số đỉnh  $n$  của đồ thị
- Nhập danh sách kề của  $n$  đỉnh
- Nhập đỉnh bắt đầu  $s$
- Nhập đỉnh kết thúc  $t$

Output:

- Đường đi từ đỉnh  $s$  đến đỉnh  $t$

Example:

Input	Output
24 1: 2 3 4 2: 1 3 4 6 3: 1 2 4 5 4: 1 2 3 7 5: 3 6 7 8 12 6: 2 5 7 12 7: 4 5 6 8 8: 5 7 12 9: 10 11 13 10: 9 11 12 13 11: 9 10 13 12: 5 6 8 10 13: 9 10 11 1 13	1 2 3 4 7 5 6 12 10 9 11 13