

HƯỚNG DẪN THỰC HÀNH THÀNH PHẦN LIÊN THÔNG

I. THUẬT TOÁN LIÊN THÔNG

Khởi tạo tất cả các đỉnh là chưa viếng thăm ($Nhan[i] = 0$, cũng dùng để lưu chỉ số của thành phần liên thông).

Khởi tạo `SoThanhPhanLT = 1`

Lặp

1. Chọn i là đỉnh của đồ thị mà chưa được viếng thăm (tức $Nhan[i] == 0$),
2. Áp dụng lan truyền bằng thuật toán tìm đường đi từ i (BFS/DFS) để xác định *một* thành phần liên thông. Trong quá trình lan truyền đánh dấu các đỉnh viếng thăm bằng `SoThanhPhanLT`
3. Tăng giá trị của `SoThanhPhanLT` lên 1 (`SoThanhPhanLT++`)

Kết thúc lặp

Để lấy thông tin từ thành phần liên thông, kiểm tra `Nhan`.

II. TÌM LIÊN THÔNG THEO CHIỀU SÂU - DFS

1. THUẬT TOÁN

Dưới đây là chi tiết thuật toán DFS xuất phát từ đỉnh s để xác định thành phần liên thông:

Viếng thăm đỉnh s

Đánh dấu đỉnh s

Lần lượt **viếng thăm** các đỉnh kề với s mà chưa được viếng thăm.

2. HƯỚNG DẪN CÀI ĐẶT

```
bool Graph::DFS(int *nhan, int v, int SoThanhPhanLT){
    // Đánh dấu đã viếng thăm v
```

```

        nhan[v] = ...;

        // Duyệt qua các đỉnh kề với đỉnh v mà chưa viếng thăm
        for (int i = 0; i < soDinh; i++){
            if(... && ...){

                // Viếng thăm đỉnh i (lưu ý thoát sớm khi tìm thấy đường
                đi)
                ViengTham(nhan, i, SoThanhPhanLT);
            }
        }
    }
}
    
```

Lưu ý rằng, sinh viên có thể dùng dùng ngăn xếp để khử đệ quy cho hàm ViengTham.

III. TÌM LIÊN THÔNG THEO CHIỀU RỘNG – BFS

1. THUẬT TOÁN

Dưới đây là chi tiết thuật toán BFS, tìm liên thông từ đỉnh s.

- 1 Khởi tạo hàng đợi *queue*
- 2 Thêm s vào hàng đợi *queue*
- 3 Trong khi *queue* $\neq \emptyset$, thực hiện{
 - v = Dequeue (*queue*)
 - Với những đỉnh i kề với v mà chưa được viếng thăm{
 - Đánh dấu viếng thăm i
 - Thêm i vào hàng đợi *queue*