

HƯỚNG DẪN CODE ĐỒ THỊ LIÊN THÔNG

Chú ý: Trong hướng dẫn này, chỗ nào có cụm từ “bạn viết code” hay đại loại thế. Thì bạn phải viết code chỗ đó hén.

Khi bạn làm tới phần này, thì bạn **đã làm được việc đọc thông tin** của đồ thị từ một file nào đó vào chương trình của bạn rồi hén. Nếu bạn vẫn chưa làm được điều này thì đề nghị bạn mở lại file “**HƯỚNG DẪN CODE NHẬP XUẤT MA TRẬN KÈ TỪ FILE**” đọc và làm. Còn nếu bạn đã làm được rồi thì chúng ta tiếp tục hén **J**

Nhắc lại: Thông tin đồ thị của bạn sẽ được lưu trữ trong chương trình thông qua một cấu trúc như sau đúng không?

```
#define MAX 10 // định nghĩa giá trị MAX
#define inputfile "C:/test.txt" // định nghĩa đường dẫn tuyệt đối đến file chứa thông tin của đồ thị
typedef struct GRAPH {
    int n; // số đỉnh của đồ thị
    int a[MAX][MAX]; // ma trận kề của đồ thị
}DOTHI;
```

Bước 1: xét tính liên thông của đồ thị, viết một hàm **DiTimCacDinhLienThong** như sau để đi tìm các đỉnh j liên thông với đỉnh i trong đồ thị và gán nhãn cho những đỉnh liên thông j đó bằng nhãn của đỉnh i.

```
void DiTimCacDinhLienThong (DOTHI g, int nhan[MAX], int i)
{
    for (int j = 0; j < g.n; j++)
```

```
{
    if (g.a[i][j] != 0 && nhan[j] != nhan[i]) // nếu tồn tại cạnh giữa đỉnh i và
đỉnh j, đồng thời nhãn của đỉnh j khác với nhãn của đỉnh i (nhãn thành phần liên thông)
thì thực hiện gán nhãn của đỉnh j = nhãn của đỉnh i và DiTimCacDinhLienThong với
đỉnh j
    {
        nhan[j] = nhan[i]; // gán nhãn cho đỉnh j
        DiTimCacDinhLienThong (g, nhan, j); // tiếp tục
DiTimCacDinhLienThong với đỉnh j và gán nhãn tương ứng tiếp.
    }
}
}
```

Sau đó viết hàm **XetLienThong** như sau:

```
void XetLienThong(DOTHI g)
{
    int Nhan[MAX]; // tạo một mảng Nhãn để lưu lại nhãn của các đỉnh trong đồ thị g
    int i;
    for (i=0; i<g.n; i++) // gán nhãn ban đầu cho tất cả các đỉnh của đồ thị g là 0
        Nhan[i] = 0;
    int SoThanhPhanLT = 0; // lưu lại số thành phần liên thông trong đồ thị g, ban đầu
là 0. Tức chưa có thành phần nào.

    // duyệt lần lượt tất cả các đỉnh và chọn đỉnh có nhãn là 0. Ta bắt đầu xét
    for (i=0; i<g.n; i++)
    {
        if (Nhan[i] == 0) // có một đỉnh trong đồ thị có nhãn là 0
        {
```

Đồ Thị Liên Thông

```
SoThanhPhanLT++; // tăng số thành phần liên thông lên
Nhan[i] = SoThanhPhanLT; // gán nhãn cho đỉnh đó bởi
SoThanhPhanLT

DiTimCacDinhLienThong(g, Nhan, i); // gọi hàm đi tìm các đỉnh
liên thông với đỉnh i và gán nhãn cho nó. Hàm này được khai báo ở sau.
    }
}

printf("So thanh phan lien thong la %d\n", SoThanhPhanLT);
/*Tới đây là coi như bạn đã hoàn tất quá trình xét tính liên thông của đồ thị rồi đó
J. Vấn đề còn lại là bạn chỉ cần code để hiển thị ra những thành phần liên thông của nó
ra. Gợi ý dựa vào mảng Nhan */
for (i = 1; i <= SoThanhPhanLT; i++)
{
    printf("Thanh phan lien thong thu %d gom cac dinh ", i);
    /* các bạn code phần này để xử lý hiển thị ra các đỉnh trong thành phần liên
thông thứ i*/
    printf("\n");
}
}
```

Bước 2: Code trong hàm main để gọi hàm các hàm tương ứng và chạy. Có thể làm như sau:

```
void main()
{
    DOTH1 g;
    clrscr();
```

```
if (DocMaTranKe("C:\\test2.txt",g) == 1)
{
    printf("Da lay thong tin do thi tu file thanh cong.\n\n");
    XuatMaTranKe(g);
    printf("Bam 1 phim bat ki de bat dau xet tinh lien thong cua do thi ...\n\n");
    getch();
    XetLienThong(g);
}
getch();
}
```

Chúc các bạn may mắn và học tốt môn này

GOOD LUCK TO U

-----HẾT-----