

HƯỚNG DẪN CODE DUYỆT THEO CHIỀU SÂU (DFS)

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.

Đề bài: Bạn có một đồ thị g (gồm đỉnh n và ma trận kề a), bạn muốn tìm đường đi từ một đỉnh S (Start) đến một đỉnh F (Finish) trong đồ thị đó. Bạn hãy viết chương trình tìm đường đi có thể theo thuật toán duyệt theo chiều sâu (DFS). Nếu tìm có đường đi từ đỉnh S (Start) đến đỉnh F (Finish) trong đồ thị g này thì bạn xuất ra đường đi, còn nếu không có đường đi thì bạn thông báo không có đường đi từ đỉnh S (Start) đến đỉnh F (Finish).

Hướng dẫn Code:

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 KÊ TỪ FILE**” đọc và làm nhé. 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: Tạo 1 mảng 1 chiều **LuuVet** dùng để lưu vết đường đi từ S à F, và 1 mảng 1 chiều khác với tên là **ChuaXet** dùng để đánh dấu đỉnh nào trong đồ thị đã xét rồi, đỉnh nào chưa xét trong quá trình tìm đường đi từ S à F.

```
int LuuVet[MAX]; // LuuVet[i] = đỉnh liền trước i trên đường đi từ S à i
```

```
int ChuaXet[MAX]; // ChuaXet[i] = 0 là đỉnh i chưa được xét đến trong quá trình tìm đường đi, còn ChuaXet[i] = 1 là đỉnh i được xét đến rồi trong quá trình tìm đường đi.
```

Bước 2: Hướng dẫn code phần duyệt theo chiều sâu (DFS), thuật toán này code theo dạng đệ qui.

```
void DFS(int v, GRAPH g) // hàm xét tại đỉnh v của đồ thị g
```

```
{
```

```
    ChuaXet[v] = 1; // gán lại giá trị đỉnh v trong mảng ChuaXet là 1, điều này có nghĩa là đỉnh v đã và đang được xét hay duyệt đến theo thuật toán.
```

```
    int u;
```

```
    for(u = 0; u < g.n ; u++)
```

```
    {
```

```
        if(g.a[v][u] != 0 && ChuaXet[u] == 0) // Xem xét có cạnh nào nối từ đỉnh v đến đỉnh u trong đồ thị g không (điều này tương ứng với g.a[v][u] != 0) và đỉnh u đã được xét hay duyệt đến hay chưa? Nếu có cạnh nối từ đỉnh v đến đỉnh u và đỉnh u chưa được xét hay duyệt đến thì tiến hành duyệt đỉnh u
```

```
        {
```

```
            LuuVet[u] = v; // đánh dấu lại đỉnh u được đi đến từ đỉnh v trong quá trình duyệt theo thuật toán DFS
```

```
            DFS(u,g); // tiến hành nhảy tới đỉnh u và xét duyệt đỉnh u
```

```
        }
```

```
    }
```

```
}
```

Duyệt theo DFS

Trước khi tiến hành duyệt DFS để tìm đường đi **S** → **F** thì cần phải khởi tạo các giá trị thích hợp cho các mảng **LuuVet** và **ChuaXet**. Quá trình đó như sau

void **duyettheoDFS** (int S, int F, GRAPH g) // **hàm này dùng để tìm đường đi từ S → F trong đồ thị g theo thuật toán DFS**

```
{
    int i;
    // khởi tạo lại các giá trị thích hợp cho mảng LuuVet và ChuaXet
    /* các bạn tự viết code khởi tạo các giá trị cho mảng LuuVet và ChuaXet nhé. Gợi ý vì ban đầu chưa chạy thuật toán nên các đỉnh i đều chưa có vết đi đến đó nên đặt giá trị -1. Và ban đầu chưa chạy thuật toán nên đỉnh i trong đồ thị g đều chưa được xét nên giá trị là 0 */

    DFS(S,g); // tiến hành thuật toán duyệt theo chiều sau
    // xuất đường đi
    if (ChuaXet[F] == 1) // sau khi thuật toán duyệt xong mà đỉnh F được xét hay duyệt đến thì nhãn của nó = 1 điều này có nghĩa là có đường đi từ S → F. Tiến hành xuất đường đi.
    {
        printf("Duong di tu dinh %d den dinh %d la: \n\t",S,F);
        i = F;
        printf("%d ", F);
        /* các bạn tự viết code xuất ra đường đi từ F → S hén. Gợi ý dựa vào mảng LuuVet để tiến hành truy vết ra đường đi từ S → F*/
    }

    Else // sau khi thuật toán duyệt xong mà đỉnh F chưa được xét hay duyệt đến thì nhãn của nó = 0 điều này có nghĩa là không có đường đi từ S → F. Thông báo không có đường đi từ S → F
```

```
{  
    printf("Khong co duong di tu dinh %d den dinh %d \n",S,F);  
}  
}
```

Bước 3: 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(inputfile, 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 duyet theo DFS ...\n\n");  
        getch();  
        duyetheoDFS(0,2,g);  
    }  
    getch();  
}
```

Chúc các bạn may mắn và học tốt môn này

GOOD LUCK TO U

-----HẾT-----