

HƯỚNG DẪN CODE TÌM CHU TRÌNH HAMILTON, ĐƯỜNG ĐI HAMILTON

Chú ý: Trong file hướng dẫn này, có những chỗ **bạn phải tự viết code**. Có những chỗ sai hoặc thiếu **bạn phải tự tìm và sửa**. Đây chỉ là file hướng dẫn không phải file code mẫu.

Khi đã làm đến bài thực hành này thì mặc định bạn đã biết việc đọc vào thông tin của một đồ thị từ một file nào đó vào chương trình của bạn. Còn nếu bạn chưa làm được điều này hoặc quên thì vui lòng dừng tại đây và xem lại bài **“hướng dẫn code nhập xuất ma trận kề từ file”** đọc và làm theo. Còn nếu nhớ rồi thì chúng ta tiếp tục.

Thông tin đồ thị của bạn sẽ được lưu trong chương trình thông qua một cấu trúc như sau:

```
#define FALSE 0    //Định nghĩa giá trị cho đỉnh chưa xét

#define TRUE 1     //Định nghĩa giá trị cho đỉnh đã xét

#define MAX 10     //Định nghĩa giá trị MAX

#define inputfile "C:/test.txt" //Đường dẫn tuyệt đối đến file chứa thông tin của đồ thị

int LuuVet[MAX];   //Mảng lưu chu trình hoặc đường đi Hamilton

int ChuaXet[MAX];  //Mảng đánh dấu các đỉnh đã thăm

int c; //Tổng số chu trình Hamilton nếu có

int d; //Tổng số đường đi Hamilton nếu có

typedef struct GRAPH

{

    int n;          //Số đỉnh của đồ thị

    int a[MAX][MAX]; //Ma trận kề của đồ thị

}DOTHI;
```

Bước 1: Ta khởi tạo một số giá trị ban đầu như sau:

//Hàm khởi tạo

```
void KhoiTao(DOTHI g)
```

```
{
```

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

```
    {
```

```
        ChuaXet[i] = TRUE; //Tất cả các đỉnh đều chưa được xét
```

```
    }
```

```
    c = d = 0; //Tổng số chu trình và đường đi Hamilton là 0
```

```
}
```

Hàm tìm chu trình Hamilton đối với một đồ thị g, hàm này viết theo kiểu đệ quy. **Chú ý tham số u trong hàm dưới đây không phải là một đỉnh của đồ thị mà là vị trí của một đỉnh trong chu trình.**

//Hàm tìm chu trình hamilton

```
void TimChuTrinhHamilton(int u, DOTHI g)
```

```
{
```

```
    //Duyệt qua tất cả các đỉnh của đồ thị
```

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

```
    {
```

```
        //Nếu đỉnh v chưa xét và có cạnh nối với đỉnh tìm được trước đó
```

```
        if(g.a[LuuVet[u - 1]][v] == 1 && ChuaXet[v] == TRUE)
```

```

{
    LuuVet[u] = v;          //Lưu lại đỉnh v vào mảng lưu vết

    ChuaXet[v] = FALSE;    //Đánh dấu đỉnh v đã được xét

    //Nếu đã đủ số đỉnh tạo nên chu trình và đỉnh v trùng với đỉnh xuất phát

    //Thì in ra chu trình này.

    /*Bạn tự thêm vào điều kiện chỗ này nha*/

    InChuTrinh(g);

    //Ngược lại, nếu chưa đủ đỉnh thì tìm đỉnh tiếp theo

    else    TimChuTrinhHamilton(u + 1, g);

    ChuaXet[v] = TRUE; //Nếu không tìm được chu trình thì gán lại đỉnh v là
chưa xét và tìm đường khác.

}

}

}

```

Tiếp đến bạn cần phải có một hàm để in ra chu trình dựa vào mảng lưu vết. Cái này bạn tự viết nhé, đơn giản với bạn quá mà.

```
void InChuTrinh(DOTHI g)
```

```

{

    //Bạn tự viết code nhé

}

```

Bước 2: Nếu đồ thị không có chu trình Hamilton thì sao? Vẫn còn khả năng là tồn tại đường đi Hamilton mà. Để kiểm tra việc này ta cần viết một hàm `TimDuongDiHamilton` như sau:

Chú ý: đây là hàm tìm đường đi từ một đỉnh xuất phát ban đầu do ta chọn, do đó tùy theo cách bạn chọn đỉnh xuất phát ban đầu sẽ quyết định kết quả thu được. Còn nếu bạn muốn in ra tất cả các đường đi thì bạn từ điều chỉnh lại code nhé.

//Hàm tìm đường đi Hamilton xuất phát từ một đỉnh

```
void TimDuongDiHamilton(int u, DOTHI g)
```

```
{
```

```
    //Duyệt tất cả các đỉnh
```

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

```
    {
```

```
        //Nếu đỉnh v chưa xét và có cạnh nối với đỉnh tìm được trước đó
```

```
        if(g.a[LuuVet[u - 1]][v] == 1 && ChuaXet[v] == TRUE)
```

```
        {
```

```
            LuuVet[u] = v; //Luu lại đỉnh v này vào mảng lưu vết
```

```
            ChuaXet[v] = FALSE; //Đỉnh v đã được xét
```

```
            //Nếu đã đủ số đỉnh tạo nên đường đi
```

```
            //=>In ra đường đi này
```

```
            if(u == g.n)    //Chỗ này đúng hay sai???
```

```
                InDuongDi(g);
```

```
            //Ngược lại, tìm đỉnh tiếp theo
```

```
            else    TimDuongDiHamilton(u + 1, g);
```

```
        }
```

```
    }
```

}

Hàm InDuongDi(DOTHI g) tương tự hàm InChuTrinh(DOTHI g). Bạn tự xử luôn nhé.

Bước 3:

Code hàm main để gọi các hàm tương ứng đã viết ở trên và chạy. Các bạn có thể tham khảo đoạn code dưới đây:

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

    DOTHI g;

    if(DocMaTranKe(inputfile, g) == 1)
    {
        XuatMaTranKe(g);

        KhoiTao(g);

        LuuVet[0] = 4;        //Giả sử xuất phát ở đỉnh số 4

        TimChuTrinhHamilton(1, g);

        if(!c)
        {
            printf("Khong co chu trinh Hamilton!\n");

            TimDuongDiHamilton(1, g);

            if(!d)    printf("Khong co duong di Hamilton!\n");
        }
    }
}
```

```

    }

    getch();
}

```

Xong, đến đây xem như chúng ta đã cơ bản hoàn thành chương trình tìm chu trình, đường đi Hamilton nếu có. Chúc các bạn thành công và học tốt môn này.