HƯỚNG DẪN CODE THUẬT TOÁN PRIM

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 KỀ 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 20 // 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: Trước khi thi hành thuật toán Prim cần phải kiểm tra đồ thị có liên thông hay không? Nếu đồ thị không liên thông thì không có cây khung nhỏ nhất. Còn ngược lại thì có.

Lật lại bài "**xét tính liên thông**", xem và code phần xét liên thông trên đồ thị. Tuy nhiên, bạn cần có một sự thay đổi nhỏ là hàm liên thông phải trả về kết quả để mình biết đường (đồ thị liên thông hay không) mà xử lý.

```
int XetLienThong(DOTHI g)
{
    /*code như phần xét liên thông chỉ thêm trả kết quả về */
```

```
        return SoThanhPhanLT;
    }
```

Bước 2: Tiến hành code thuật toán **Prim** như sau:

```
int ChuaXet[MAX]; // gia tri 0 la chua xet, gia tri 1 la xet roi
typedef struct EDGE // khai báo 1 cấu trúc CANH cho cạnh của đồ thị
{
    int u;
    int v;
    int value;
}CANH;

CANH T[MAX]; // mảng lưu các cạnh trong thuật toán Prim
void Prim (DOTHI g)
{
    if (XetLienThong(g) != 1) // đi kiểm tra đồ thị có liên thông không, nếu kết quả trả về là 1 thì đồ thị liên thông (tức chỉ có 1 thành phần liên thông), còn khác 1 thì đồ thị không liên thông
    {
        printf ("Do thi khong lien thong, do do khong thuc hien duoc thuat toan Prim tim cay khung nho nhat\n");
        return;
    }
    int nT = 0; // dùng để lưu số cạnh trong thuật toán Prim
    for (int i = 0; i < g.n; i++) // ban đầu thuật toán Prim, chưa làm gì nên gán giá trị chưa xét cho tất cả các đỉnh bằng 0
        ChuaXet[i] = 0;
    ChuaXet[0] = 1; // bắt đầu thuật toán Prim, xuất phát từ đỉnh 1
    while (nT < g.n - 1) // nếu thuật toán đã thu thập đủ g.n-1 cạnh thì thuật toán dừng
```

```
{
        CANH CanhNhoNhat; // dùng để lưu cạnh nhỏ nhất nối từ tập những đỉnh đã xét (giá trị chưa xét == 1) đến 1 đỉnh chưa xét nào đó trong đồ thị.
        int GiaTriNhoNhat = 100; // khởi tạo giá trị nhỏ nhất của cạnh nhỏ nhất là 100, bạn có thể thay đổi số này sao cho phù hợp với bài toán của bạn.
        for (int i = 0; i < g.n; i++) // duyệt các đỉnh trong đồ thị
        {
                if (ChuaXet[i] == 1) // xem đỉnh nào đã xét
                {
                        for (int j = 0; j < g.n; j++)
                                if (ChuaXet[j] == 0 && g.a[i][j] != 0 && GiaTriNhoNhat > g.a[i][j]) // tìm đỉnh chưa xét j, có cạnh nối từ đỉnh i đến đỉnh j và giá trị của cạnh đó nhỏ hơn giá trị nhỏ nhất ở trên
                                {
                                        CanhNhoNhat.u = i; // cập nhập lại giá trị trong CanhNhoNhat
                                        CanhNhoNhat.v = j;
                                        CanhNhoNhat.value = g.a[i][j];
                                        GiaTriNhoNhat = g.a[i][j]; // cập nhập lại GiaTriNhoNhat
                                }
                }
        }
        T[nT] = CanhNhoNhat; //Thêm cạnh nhỏ nhất vào tập cạnh T của mình
        nT++; // tăng số cạnh lên 1
        ChuaXet[CanhNhoNhat.v] = 1; // gán lại giá trị chưa xét của đỉnh v là 1, tức đã xét rồi
}
```

```
	int TongTrongSoCuaCayKhung = 0; // trọng số của cây khung nhỏ nhất
	// xuất cây khung nhỏ nhất bởi thuật toán Prim và giá trị của cây khung
	printf ("Cay khung nho nhat cua do thi la \n");
	for (i = 0; i < nT - 1; i++)
	{
		printf ("(%d,%d), ", T[i].u, T[i].v);
		TongTrongSoCuaCayKhung += T[i].value;
	}
	printf ("(%d,%d).\n", T[nT - 1].u, T[nT - 1].v);
	TongTrongSoCuaCayKhung += T[nT - 1].value;
	printf ("Tong gia tri cua cay khung la %d\n",TongTrongSoCuaCayKhung);
}
```

Bước 3: Code trong hàm main để gọi hàm các hàm tương ứng và chạy. Có thể làm như sau:

```
void main()
{
	DOTHI 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 tim cay khung nho nhat ...\n\n");
		getch();
		Prim(g);
	}
	getch();
}
```

HƯỚNG DẪN CHI TIẾT TỚI CỠ NÀY RỒI MÀ BẠN VẪN KHÔNG LÀM ĐƯỢC NỮA THÌ BẠN CHUẨN BỊ TINH THẦN ĐI HÉN J

Chúc các bạn may mắn và học tốt môn này

GOOD LUCK TO U