

LAB 5

Lưu ý:

- Làm bài vào file word bằng tiếng Việt hoặc tiếng Anh
- Những bài làm giống nhau sẽ bị 0 điểm
- Với những bài lập trình, cần phải copy mã nguồn và chụp màn hình kết quả, đưa vào file word
- Địa chỉ nộp bài: ctdlgt.bku@gmail.com
- Hạn chót nộp bài: 23 giờ ngày 10/8/2023

Sinh viên đọc hiểu và hoàn thiện chương trình sau nguồn sau. Chương trình này đọc dữ liệu đồ thị từ file văn bản, sau đó thực hiện một số thao tác trên đồ thị.

```
#include <iostream>
#include <stdio.h>
#include <math.h>
#include <string.h>

using namespace std;

#define      MAX_VERTEX      50

#define      UNDIRECTEDGRAPH      0
#define      DIRECTEDGRAPH      1

#define      DISPLAYGRAPH      0
#define      DEPTHFIRSTTRAVERSAL      1

#define      INCREASINGORDER      0
#define      DECREASINGORDER      1

struct VertexType{
    char      strName[10];
    float      weight;
    bool      isMarked;
    VertexType *next;
    VertexType(){
        strcpy(strName, "");
        weight = 0;
        isMarked = false;
        next = NULL;
    }
};

struct GraphNode{
    VertexType vertex;
    VertexType *adjVertex;
    int      indegree;
    int      outdegree;
    bool      isMarked;
    GraphNode(){
        indegree = 0;
        outdegree = 0;
        isMarked = false;
        adjVertex = NULL;
    }
}
```

```

};

struct Graph{
    int          nGraphType;
    int          nOperation;
    int          nVertexNum;
    VertexType   startVertex;
    int          nOrder;

    GraphNode    graph[MAX_VERTEX];
};

Graph    myGraph;

int readFile(Graph &myGraph, char* filename ){
    return 1;
}

void DisplayGraph(Graph &myGraph){

}

string depthFirstTraversal(Graph &myGraph, VertexType startVertex){
    string result="";

    return result;
}

int main()
{
    for(int i = 0; i<MAX_VERTEX; i++)
        strcpy(myGraph.graph[i].vertex.strName, "-1");

    readFile(myGraph, "input.txt");

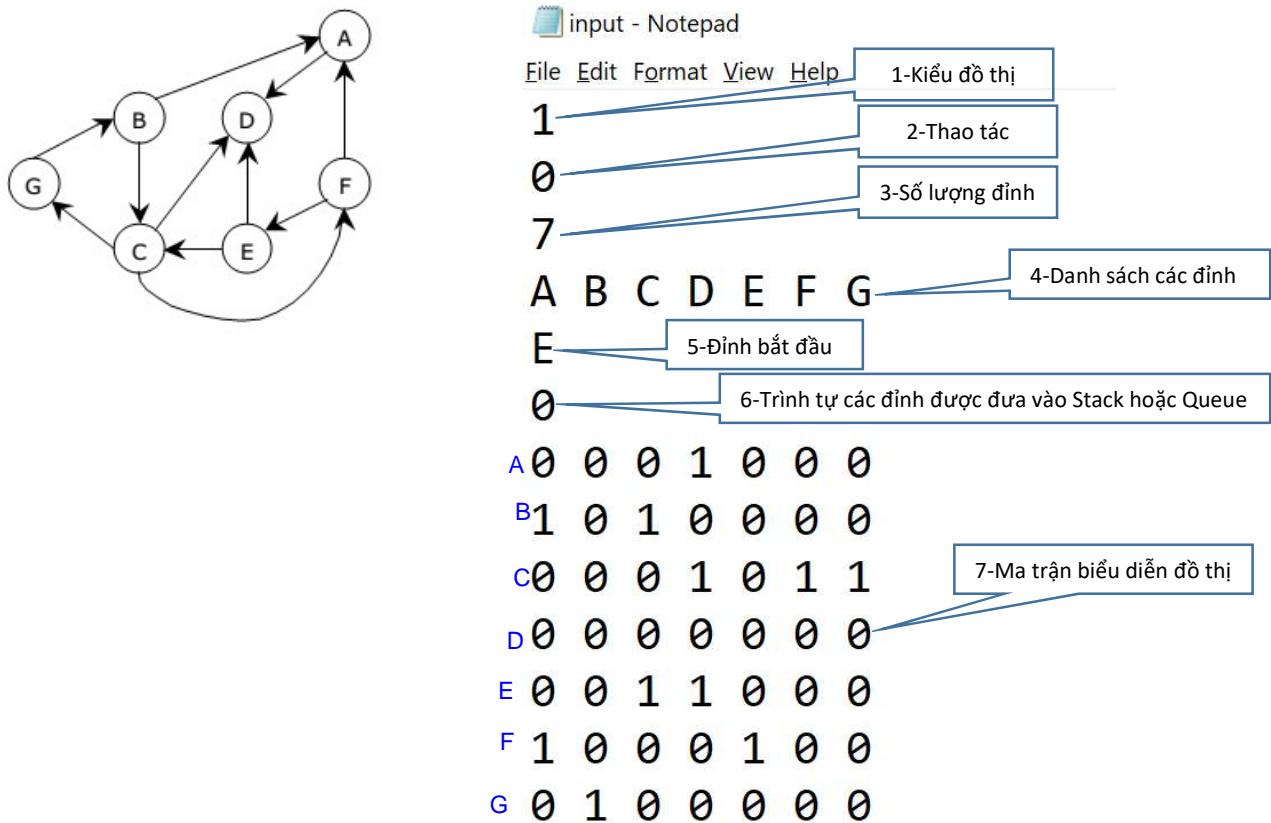
    switch(myGraph.nOperation)
    {
    case DISPLAYGRAPH:
        cout << "Display Graph" << endl;
        DisplayGraph(myGraph);
        cout << "-----" << endl;
        break;
    case DEPTHFIRSTTRAVERSAL:
        cout << "Depth First Traversal" << endl;
        cout<< depthFirstTraversal(myGraph,myGraph.startVertex)<<endl;
        cout << "-----" << endl;
        break;
    }

    return 0;
}

```

I - ĐỊNH DẠNG FILE DỮ LIỆU

Hình 1 là một ví dụ về đồ thị và file dữ liệu tương ứng của nó



Hình 1

1) Kiểu đồ thị

Giá trị này bằng 0 → đồ thị vô hướng

Giá trị này bằng 1 → đồ thị có hướng

2) Thao tác

Giá trị này bằng 0 → hiển thị đồ thị dưới dạng danh sách kề (adjacency list)

Giá trị này bằng 1 → duyệt đồ thị theo chiều sâu

3) Số lượng đỉnh

Giá trị này cho biết đồ thị có tổng cộng bao nhiêu đỉnh

4) Danh sách các đỉnh

Liệt kê tên tất cả các đỉnh của đồ thị

5) Đỉnh bắt đầu

Với một số thao tác, cần phải chỉ định đỉnh bắt đầu, chẳng hạn như thao tác duyệt đồ thị

6) Trình tự đưa các đỉnh vào Stack hoặc Queue

Một số thao tác như duyệt đồ thị hoặc tìm thứ tự topo, cần phải chỉ định trình tự đưa các đỉnh vào Stack hoặc Queue.

Giá trị này bằng 0 → Các đỉnh được đưa vào theo thứ tự từ bé đến lớn (xếp theo bảng chữ cái)

Giá trị này bằng 1 → Các đỉnh được đưa vào theo thứ tự từ lớn đến bé (xếp theo bảng chữ cái)

7) Ma trận biểu diễn đồ thị

Đối với ma trận không có trọng số thì dùng giá trị 0 và 1, nếu có trọng số thì sử dụng trọng số.

II – YÊU CẦU

Sinh viên hoàn thiện các hàm trong chương trình khởi động, có thể viết thêm một số hàm phụ trợ.

1) Hàm `int readFile(Graph &myGraph, char* filename)`

Hàm này có chức năng đọc file dữ liệu vào trong cấu trúc `myGraph`. Sinh viên tham khảo bài tập lớn 1 để biết cách đọc file văn bản.

2) Hàm `void DisplayGraph(Graph &myGraph)`

Hàm này hiển thị đồ thị dưới dạng danh sách kề. Ví dụ: với file dữ liệu giống như *Hình 1*, thì chương trình sẽ xuất ra kết quả như sau:

```
Display Graph
A: D
B: A C
C: D F G
D:
E: C D
F: A E
G: B
-----
```

3) Hàm `string depthFirstTraversal(Graph &myGraph, VertexType startVertex)`

Hàm này trả về kết quả duyệt đồ thị theo chiều sâu.