|  |
| --- |
| **TRƯỜNG ĐẠI HỌC ĐÀ LẠT**  **KHOA CÔNG NGHỆ THÔNG TIN**    **BÁO CÁO MÔN HỌC**  **CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT 2**  **Nhóm 10**  **Tên đề tài:**  **ĐỒ THỊ BIỂU DIỄN BẰNG DANH SÁCH CẠNH**          **Giảng viên hướng dẫn: Đinh Viết Tuấn**  **Sinh viên thực hiện: 1610207 – La Quốc Thắng**  **1610229 – Cao Quốc Bảo Toàn**  **1610235 – Nguyễn Hiếu Trung**  ***Đà Lạt – 04/2018*** |

**Mục lục**

[I. Tổng quan về phương pháp 4](#_Toc509251726)

[II. Mục tiêu 5](#_Toc509251727)

[III. Cài đặt 5](#_Toc509251728)

[IV. Các thao tác trên cấu trúc dữ liệu 7](#_Toc509251729)

[1. Thêm một đỉnh có nhãn label vào đồ thị 7](#_Toc509251730)

[2. Xuất thông tin của một đỉnh trong đồ thị 7](#_Toc509251731)

[3. Kiểm tra hai đỉnh u, v có kề nhau (có cạnh nối giữa chúng) hay không? 7](#_Toc509251732)

[4. Thêm một cạnh có trọng số w nối 2 đỉnh u, v vào đồ thị 7](#_Toc509251733)

[5. Lưu thông tin đồ thị xuống file 7](#_Toc509251734)

[6. Tạo đồ thị dữ liệu được lấy từ file 7](#_Toc509251735)

[7. Duyệt đồ thị theo chiều rộng 7](#_Toc509251736)

[8. Duyệt đồ thị theo chiều sâu 7](#_Toc509251737)

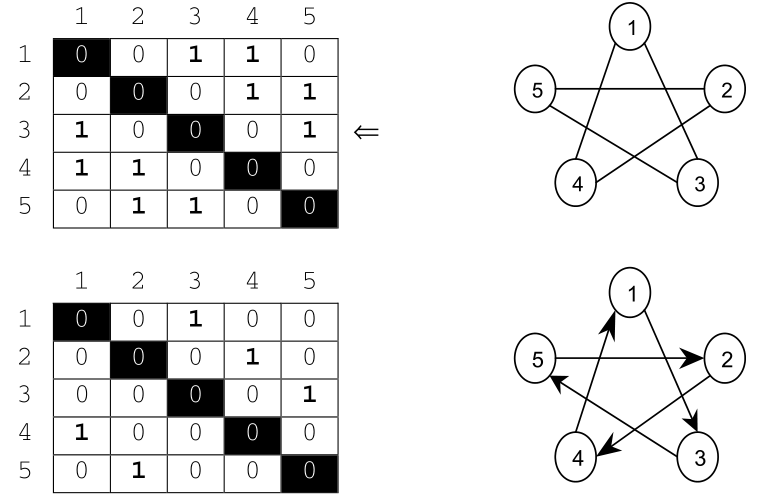
[V. Kết luận 7](#_Toc509251738)

[1. Nhận xét 8](#_Toc509251739)

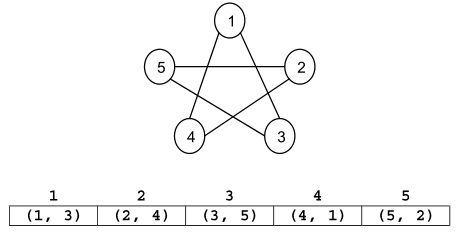
[2. Hướng phát triển 8](#_Toc509251740)

[3. Hạn chế - Khó khăn 8](#_Toc509251741)

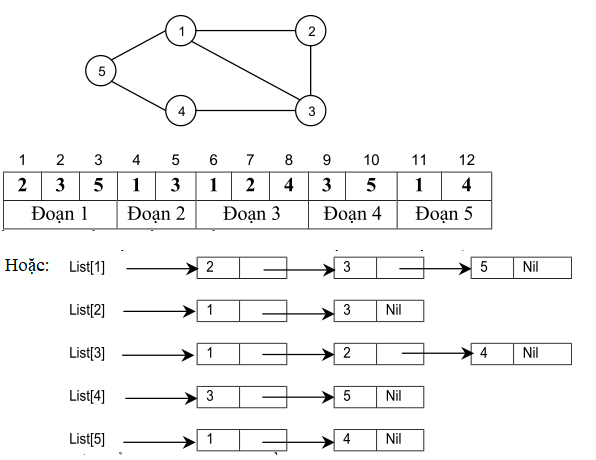
# Tổng quan về phương pháp

Để lưu trữ đồ thị và thực hiện các thuật toán khác nhau với đồ thị trên máy tính, người ta thường dùng những cấu trúc sau để biểu diễn như ma trận liền kề (ma trận kề), danh sách cạnh, danh sách kề,…

*Hình 1. Minh họa bằng ma trận liền kề (ma trận kề)*



*Hình 2. Minh họa bằng danh sách cạnh*

*Hình 3. Minh họa bằng danh sách kề*

Trong trường hợp đồ thị có n đỉnh, m cạnh, ta có thể biểu diễn đồ thị dưới dạng danh sách cạnh, trong cách biểu diễn này, người ta liệt kê tất cả các cạnh của đồ thị trong một danh sách, mỗi phần tử của danh sách là một cặp (u, v) tương ứng với một cạnh của đồ thị. (Trong trường hợp đồ thị có hướng thì mỗi cặp (u, v) tương ứng với một cung, u là đỉnh đầu và v là đỉnh cuối của cung)

# Mục tiêu

# Cài đặt

Về mặt phương tiện, ta sẽ sử dụng danh sách liên kết đơn lưu trữ các cạnh có trong đồ thị.

Trong dự án này, ta tạo ra 3 tập tin nằm trong thư mục Header Files như sau:

* Menu.h: Định nghĩa menu vận hành các chức năng của chương trình.
* Common.h: Định nghĩa các hằng số và kiểu dữ liệu của đồ thị.
* Graph.h: Định nghĩa các thao tác trên đồ thị.

Thư mục Resource Files ta tạo 1 tập tin Data.txt chứa dữ liệu về đồ thị có cấu trúc xác định để việc đọc dữ liệu thành công.

Thư mục cuối cùng là Source Files, tạo 1 tập tin là Program.cpp chứa hàm main để gọi hàm XuLyMenu ở file Menu.h thực thi các chức năng.

Các bước tiến hành cài đặt:

1. Trong tập tin Common.h:

* Định nghĩa các hằng số và kiểu dữ liệu:

#define UPPER 100 //Số phần tử tối đa

#define ZERO 0 //Giá trị 0

#define MAX 20 //Số đỉnh tối đa

#define INF 1000 //Vô cùng

#define YES 1 //Đã xét

#define NO 0 //Chưa xét

#define NULLDATA -1 //Giá trị rỗng

* Định nghĩa các kiểu dữ liệu:

typedef char LabelType; //Kiểu tên nhãn là một ký tự

typedef int CostType; //Trọng số

* Định nghĩa cấu trúc một cạnh:

struct Edge

{

int Marked; //Trạng thái

char Target; //Đỉnh cuối

CostType Weight; //Trọng số

Edge\* Next; //Cạnh tiếp

};

* Định nghĩa cấu trúc của một đỉnh:

struct Vertex

{

LabelType Label; //Nhãn của đỉnh

int Visited; //Trạng thái

Edge\* EdgeList; //DS cạnh kề

};

* Định nghĩa một đường:

struct Path

{

CostType Length; //Độ dài đường đi

int Parent; //Đỉnh trước

};

* Định nghĩa danh sách cạnh:

typedef Edge\* EdgePtr;

* Định nghĩa kiểu dữ liệu đồ thị:

struct Graph

{

bool Directed; //ĐT có hướng hay không

int NumVertices; //Số đỉnh

int NumEdges; //Số cạnh

EdgePtr Edges[MAX]; //DS các cạnh

};

1. Trong tập tin Program.cpp:

#include <iostream>

#include <conio.h>

#include <fstream>

#include <stack>

#include <queue>

using namespace std;

#include "Common.h"

#include "Graph.h"

#include "Menu.h"

int main()

{

int soMenu = 8, menu;

do

{

menu = ChonMenu(soMenu);

XuLyMenu(menu);

} while (menu > 0);

return 1;

}

# Các thao tác trên cấu trúc dữ liệu

## Thêm một đỉnh có nhãn label vào đồ thị

## Xuất thông tin của một đỉnh trong đồ thị

## Kiểm tra hai đỉnh u, v có kề nhau (có cạnh nối giữa chúng) hay không?

## Thêm một cạnh có trọng số w nối 2 đỉnh u, v vào đồ thị

## Lưu thông tin đồ thị xuống file

## Tạo đồ thị dữ liệu được lấy từ file

## Duyệt đồ thị theo chiều rộng

## Duyệt đồ thị theo chiều sâu

# Kết luận

## Nhận xét

## Hướng phát triển

## Hạn chế - Khó khăn

Nhược điểm cơ bản của danh sách cạnh là khi ta cần duyệt tất cả các đỉnh kề với đỉnh v nào đó của đồ thị, thì chẳng có cách nào khác là phải duyệt tất cả các cạnh, lọc ra những cạnh có chứa đỉnh v và xét đỉnh còn lại. Điều đó khá tốn thời gian trong trường hợp đồ thị dày (nhiều cạnh).