**BÁO CÁO ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**TUẦN 1**

## Tiến độ nhiệm vụ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tiến trình hoàn thành môn Đồ án CNTT | | | | | | | | | | | | | | | | |
|  | Các phương thức | TKB | | | | | | | | | | | | | Cường | Quang |
| 1 | Tạo 1 đồ thị | o |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Thêm một đỉnh vào đồ thị đã có |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Thay đổi thông tin của một đỉnh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Thêm cạnh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Thay đổi trọng số của cạnh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Xuất các tên đỉnh, tên cạnh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Xuất thông tin 1 đỉnh, 1 cạnh | o |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Xuất ma trận kề, ma trận liên kết |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Duyệt đồ thị theo chiều rộng, chiều xâu |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Đường đi ngắn nhất từ đỉnh v đến đỉnh w |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Kiểm tra có chu trình Euler |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Cây khung bé nhất |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Day | | 03/09/2023 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Week | | 01 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Note | |  |  |  |  |  |  |  |  |  | o – Begin  o – Complete 50%  o – Complete 100% | | | | | |

## Định nghĩa về đồ thị

Lý thuyết đồ thị là nghiên cứu về điểm và đường. Trong Toán học, nó là một lĩnh vực phụ liên quan đến việc nghiên cứu các đồ thị. Nó là một biểu diễn bằng hình ảnh đại diện cho sự thật Toán học. Lý thuyết đồ thị là nghiên cứu về mối quan hệ giữa các đỉnh (nút) và các cạnh (đường).

Về mặt hình thức, một đồ thị được biểu thị là một cặp G (V, E).

Trong đó V đại diện cho các đỉnh của tập hữu hạn và E đại diện cho các cạnh của tập hữu hạn.

Do đó, chúng ta có thể nói một đồ thị bao gồm tập các đỉnh V và tập các cạnh E. không rỗng.

### Ví dụ

Giả sử, một Đồ thị G = (V, E), trong đó

Các đỉnh, V = {a, b, c, d}

Các cạnh, E = {{a, b}, {a, c}, {b, c}, {c, d}}

## Thiết kế

1. **Tạo đồ thị**

Khai báo thư viện <iostream> và <fstream> trong IDE Dev-C++. Trong đó thư viện <fstream> để nhập dữ liệu đầu vào bằng file thay vì người sử dụng tự nhập để dễ dàng, tiện lợi hơn trong quá trình sử dụng ứng dụng.

#include <iostream>

#include <fstream>

Tạo hàm createGraph()

void createGraph() {

do {

cout << "Chon cach nhap ma tran ke: " << endl;

cout << "1.Nhap bang tay" << endl;

cout << "2.Nhap bang file" << endl;

cin >> chooseGraph;

}while(chooseGraph !=1 && chooseGraph !=2);

if(chooseGraph==1) {

input(a,n);

}

else {

inputFile(a,n);

}

}

Trong hàm createGraph() sẽ có 2 lựa chọn để người dụng có thể chọn. Lựa chọn đầu tiên là nhập bằng tay, lựa chọn thứ hai sẽ import từ file có sẵn trong máy tính.

Với lựa chọn đầu tiên là nhập bằng tay, tạo ra hàm input().

void input(int a[][100],int &n) {

do {

cout << "Nhap so luong dinh cua do thi : ";

cin >>n;

}while(n<=0||n>100);

cout << "Nhap " << n << " ten cac dinh cua do thi theo thu tu : ";

for(int i=0;i<n;i++)

{

cin >> vertex[i];

}

//nhap ma tran

for(int i=0;i<n;i++)

{

cout << "Nhap tinh chat ke cua cac dinh voi dinh " << vertex[i] << " co(1), khong(0) : ";

for(int j=0;j<n;j++)

{

cin >> a[i][j];

}

}

}

Với lựa chọn thứ hai là import từ file bên ngoài, tạo ra hàm inputFile ().

void inputFile(int a[][100],int &n) {

ifstream iFile;

iFile.open("lienketke.txt");

if(iFile.is\_open()) {

iFile >>n;

for(int i = 0; i < n; i++) {

iFile >> vertex[i];

}

for(int i = 0; i < n; i++) {

for(int j=0; j < n ; j++) {

iFile >> a[i][j];

}

}

iFile.close();

cout << "Doc file thanh cong !!\n";

}

else {

cout << "Loi, khong mo duoc file !!\n";

}

}

void input(int a[][100],int &n) {

do {

cout << "Nhap so luong dinh cua do thi : ";

cin >>n;

}while(n<=0||n>100);

cout << "Nhap " << n << " ten cac dinh cua do thi theo thu tu : ";

for(int i=0;i<n;i++)

{

cin >> vertex[i];

}

//nhap ma tran

for(int i=0;i<n;i++)

{

cout << "Nhap tinh chat ke cua cac dinh voi dinh " << vertex[i] << " co(1), khong(0) : ";

for(int j=0;j<n;j++)

{

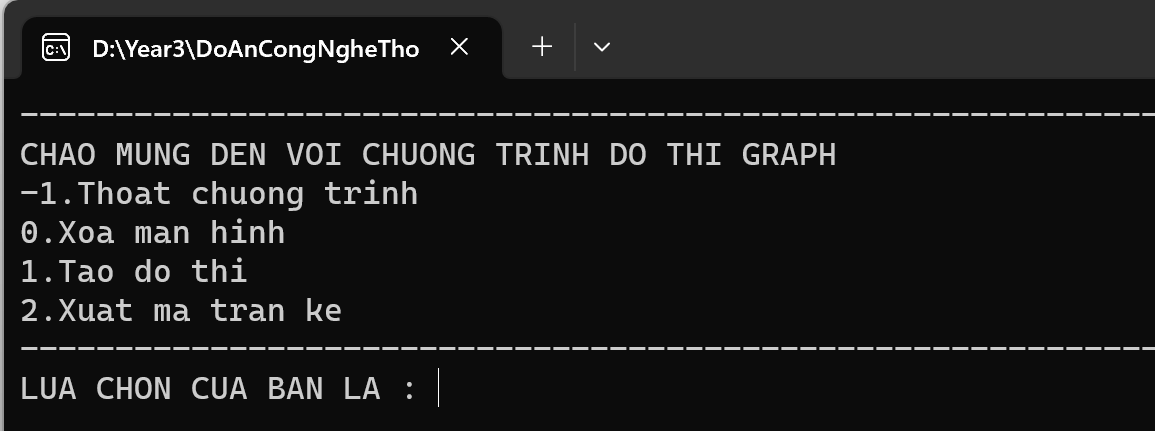
cin >> a[i][j];

}

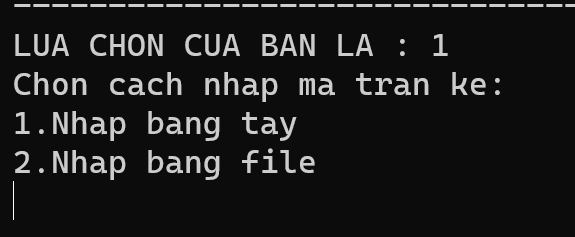
}

}

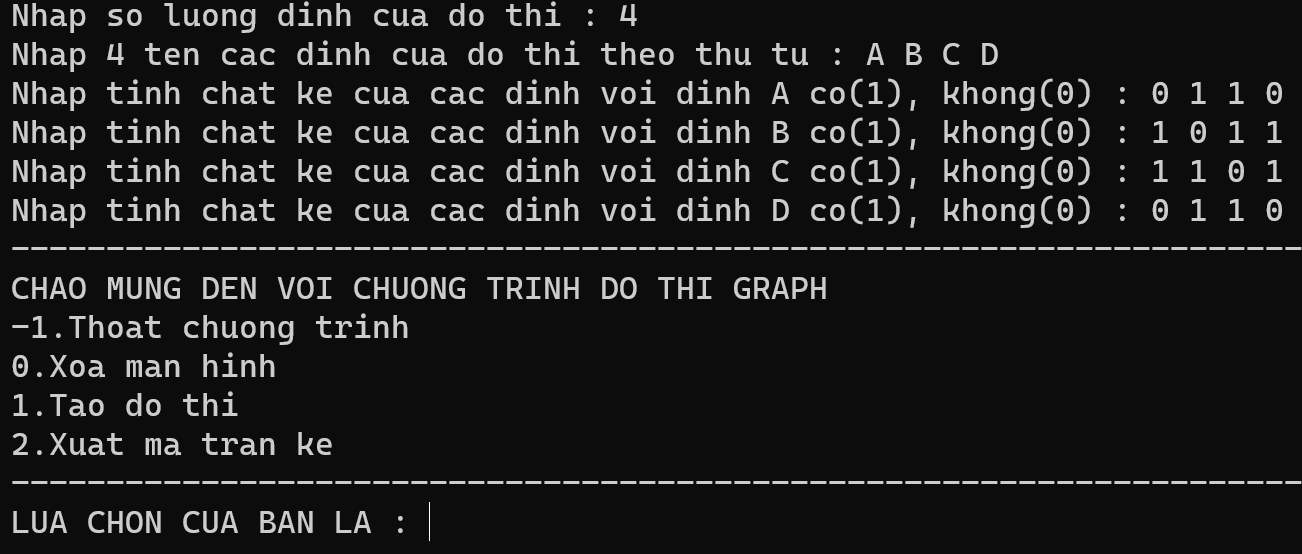
Kết quả khi chạy chương trình :



Khi chọn số 1:



Chọn tiếp số 1 để nhập tay:



Khi đã nhập xong dữ liệu. Đồ thị sẽ được lưu và chương trình sẽ tiếp tục hỏi lại chức năng bạn muốn sử dụng.

1. **Xuất đồ thị**

Sau khi đã nhập xong, tạo hàm xuất ra đồ thị ma trận kề là output().

void output(int a[][100],int n) {

for(int i=0;i<n;i++) cout << "----------";

cout << endl;

cout << "|";

cout << "\t";

for(int i=0;i<n;i++) {

cout << "|" << vertex[i] << "\t";

}

cout << "|" << endl;

for(int i=0;i<n;i++) {

cout << "|" << vertex[i] << "\t";

for(int j=0;j<n;j++) {

cout << "|" << a[i][j] << "\t";

}

cout << "|" << endl;

}

for(int i=0;i<n;i++) cout << "----------";

}

Tạo hàm Begin() để khởi tạo chương trình.

int Begin() {

cout << "------------------------------------------------------------------------------------------------------------------------\n";

cout << "CHAO MUNG DEN VOI CHUONG TRINH DO THI GRAPH\n";

cout << "-1.Thoat chuong trinh\n";

cout << "0.Xoa man hinh\n";

cout << "1.Tao do thi\n";

cout << "2.Xuat ma tran ke\n";

cout << "------------------------------------------------------------------------------------------------------------------------\n";

cout << "LUA CHON CUA BAN LA : ";

cin >> chooseBegin;

return chooseBegin;

}

Cuối cùng liên kết các hàm đó vào hàm main().

int main() {

chooseBegin = Begin();

switch(chooseBegin) {

case -1 : return 0;

case 0 : system("cls");

main();

break;

case 1 : createGraph();

main();

break;

case 2 : output(a,n);

main();

break;

default : cout << "Vui long chon so hop le !!!\n";

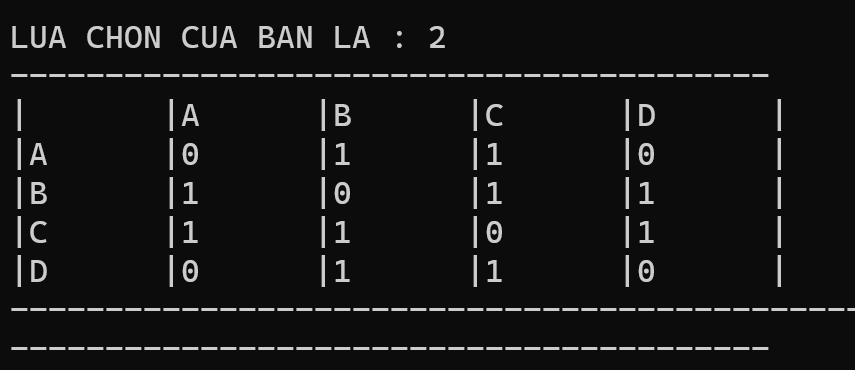
main();

}

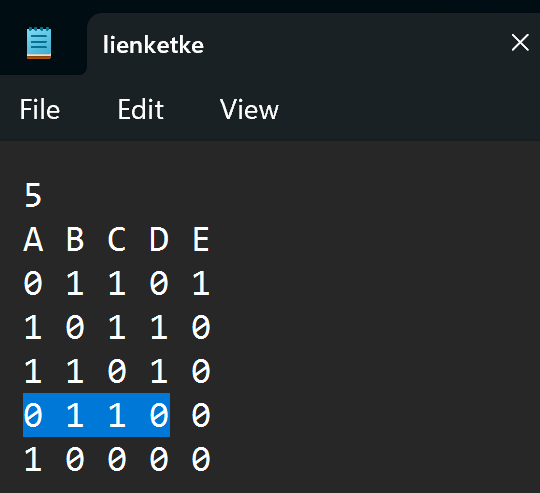
return 0;

}

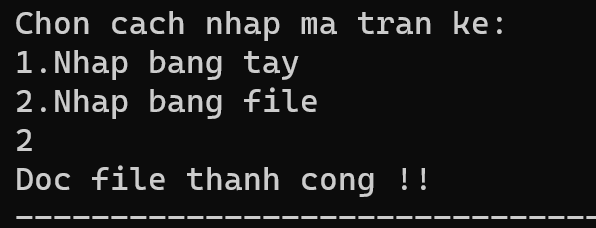
Nếu chọn 2 để xuất ma trận kề:



Dữ liệu file lienketke.txt:



Khi muốn nhập đồ thị khác bằng file:



Và kết quả khi xuất ma trận kề thay đổi:

