**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**



**BÁO CÁO ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**TUẦN 2**

***ĐỀ TÀI:* GRAPH**

**GVHD**: ThS. Nguyễn Quang Ngọc

**Sinh viên thực hiện**:

|  |  |
| --- | --- |
| **HỌ VÀ TÊN** | **MÃ SỐ SINH VIÊN** |
| Nguyễn Minh Quang | 20143481 |
| Đỗ Minh Cường | 21110147 |

## Tiến độ nhiệm vụ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tiến trình hoàn thành môn Đồ án CNTT | | | | | | | | | | | | | | | | |
|  | Các phương thức | TKB | | | | | | | | | | | | | Cường | Quang |
| 1 | Tạo 1 đồ thị | o | o |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Thêm một đỉnh vào đồ thị đã có |  |  | o |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Thay đổi thông tin của một đỉnh |  |  | o |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Thêm cạnh |  |  |  | o |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Thay đổi trọng số của cạnh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Xuất các tên đỉnh, tên cạnh |  |  |  | o |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Xuất thông tin 1 đỉnh, 1 cạnh | o | 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 | 10/09/2023 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Week | | 01 | 02 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Note | |  |  |  |  |  |  |  |  |  | o – Begin  o – Complete 50%  o – Complete 100% | | | | | |

## Định nghĩa về ma trận kề, ma trận liên kết, ma trận trọng số,danh sách liên kết:

1. **Ma trận Kề (Adjacency Matrix)**:
   * Ma trận kề là một ma trận vuông có kích thước **n x n**, trong đó **n** là số đỉnh của đồ thị.
   * Mỗi phần tử **A[i][j]** của ma trận kề có giá trị **1** nếu có cạnh nối giữa đỉnh **i** và đỉnh **j**, và **0** nếu không có cạnh nối.
   * Ma trận kề thường được sử dụng để biểu diễn đồ thị vô hướng (không có hướng).

Ví dụ:

A B C D

A 0 1 1 0

B 1 0 0 1

C 1 0 0 1

D 0 1 1 0

Trong ma trận này, có cạnh nối giữa đỉnh A và B, A và C, B và D, C và D, nhưng không có cạnh nối giữa A và D hoặc B và C.

1. **Ma trận Liên Kết (Incidence Matrix)**:
   * Ma trận liên kết là một ma trận có kích thước **n x m**, trong đó **n** là số đỉnh và **m** là số cạnh của đồ thị.
   * Mỗi phần tử **B[i][j]** của ma trận liên kết có giá trị **1** nếu đỉnh **i** tham gia vào cạnh **j**, **-1** nếu đỉnh **i** là đỉnh cuối của cạnh **j**, và **0** nếu đỉnh **i** không tham gia vào cạnh **j**.
   * Ma trận liên kết thường được sử dụng để biểu diễn đồ thị có hướng.

**VD:**  Giả sử ta có đồ thị:

A - - B

|| ||

D - - C

Đồ thị được biểu diễn bằng ma trận liên kết như sau:

A B C D

A 0 1 0 1

B 1 0 1 0

C 0 1 0 1

D 1 0 1 0

1. **Ma trận Trọng Số (Weighted Matrix)**:
   * Ma trận trọng số là một biến thể của ma trận kề hoặc ma trận liên kết, trong đó mỗi phần tử **W[i][j]** của ma trận chứa trọng số của cạnh giữa đỉnh **i** và đỉnh **j**.
   * Ma trận trọng số thường được sử dụng khi cần biểu diễn trọng số của các cạnh trong đồ thị, chẳng hạn như đồ thị đường đi ngắn nhất.

Ví dụ:

A B C D

A 0 2 3 0

B 0 0 0 0

C 0 0 0 4

D 0 0 0 0

Mỗi dòng và cột tương ứng với một đỉnh.

Giá trị tại vị trí **W[i][j]** trong ma trận trọng số là trọng số của cạnh từ đỉnh **i** đến đỉnh **j.**

1. **Thay đổi thông tin một đỉnh**

Trong lý thuyết đồ thị, việc "thay đổi thông tin của một đỉnh" đề cập đến việc sửa đổi dữ liệu hoặc thuộc tính của một đỉnh cụ thể trong đồ thị. Điều này có thể liên quan đến việc thay đổi tên, trạng thái, hoặc bất kỳ thông tin nào khác mà ta muốn liên kết với đỉnh đó.

1. **Danh sách liên kết**

Một ****danh sách liên kết (Linked List)**** là một cấu trúc dữ liệu tuyến tính, bao gồm một chuỗi các node kết nối với nhau. Mỗi node có thể xem như một phần tử trong danh sách. Mỗi node sẽ lưu trữ dữ liệu (data) của node đó và địa chỉ (address) của node kế tiếp. Bên dưới là minh họa cấu trúc dữ liệu danh sách liên kết.

[](https://gochocit.com/wp-content/uploads/2021/10/singly-linked-list.webp)

Danh sách liên kết phải có node bắt đầu, được gọi là HEAD. Node cuối cùng được gọi là TAIL. Mỗi node có nhiều nhất 1 phần tử đứng trước cũng như có nhiều nhất 1 phần tử đứng sau.

Có nhiều trường hợp trong thực tế có thể được lưu trữ bằng cách sử dụng Linked List như:

* + Danh sách học sinh
  + Danh mục sách trong thư viện
  + Danh bạ điện thoại
  + Danh sách các nhân viên trong công ty
  + …

## Thiết kế

**1. Thêm một đỉnh vào đồ thị đã có**

TTạo hàm addvertex().

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

if(n<100) {

n++;

cout << "Nhap ten dinh moi: ";

cin >> vertex[n-1];

//nhap ma tran

cout << "Nhap tinh chat ke cua dinh " << vertex[n-1] << " voi dinh khac (co(1), khong(0)): \n";

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

{

cout << "Voi dinh " << vertex[i] << " :";

cin >> a[i][n-1];

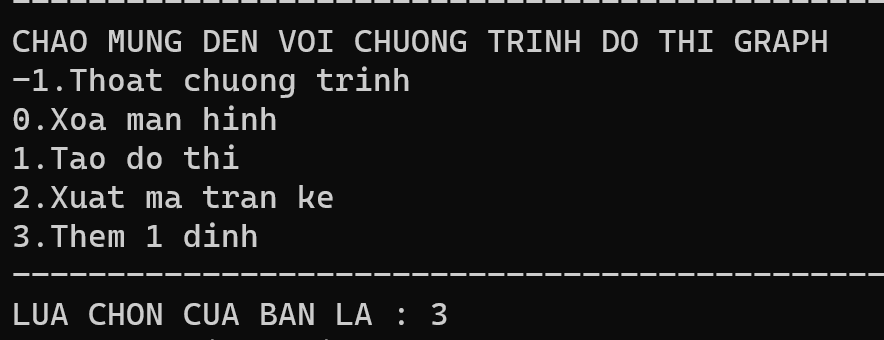
a[n-1][i] = a[i][n-1];

}

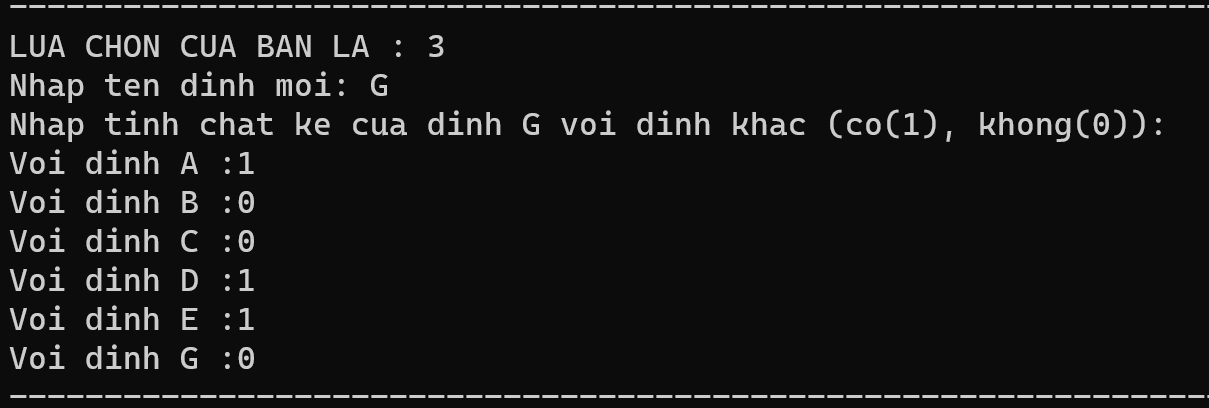
}

}

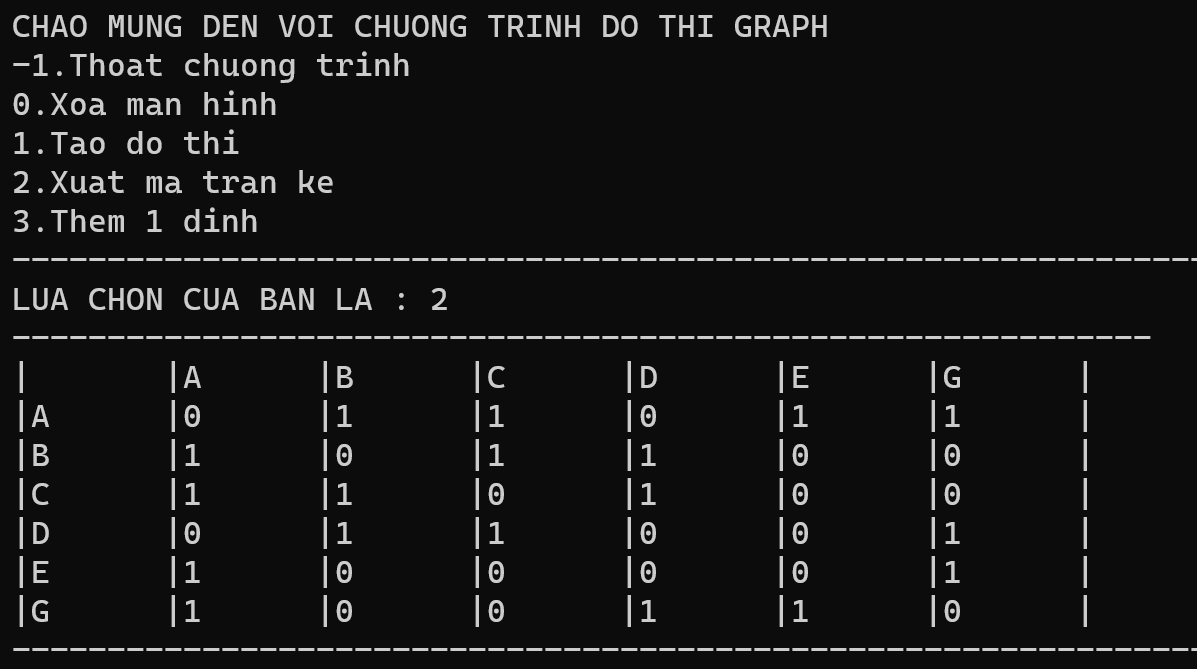
Kết quả khi chạy chương trình :



Khi chọn lựa số 3 và nhập các dữ liệu đầu vào :



Kết quả khi xuất ma trận:

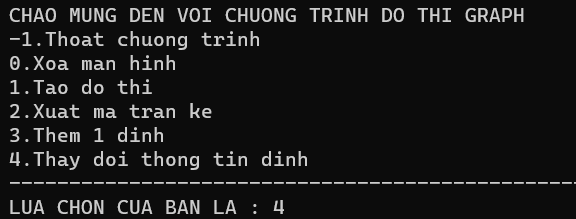


1. **Thay đổi thông tin của 1 đỉnh:**

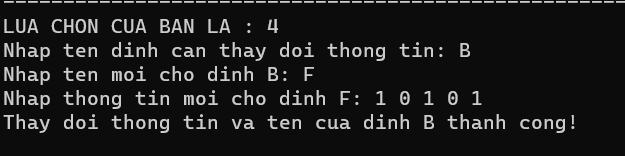
Tạo hàm changeVertexInfo:

|  |
| --- |
| void changeVertexInfo(int a[][100], int n)  {  char oldVertexName;  cout << "Nhap ten dinh can thay doi thong tin: ";  cin >> oldVertexName;  int vertexIndex = -1;  for (int i = 0; i < n; i++) {  if (vertex[i] == oldVertexName) {  vertexIndex = i;  break;  }  }  if (vertexIndex == -1) {  cout << "Dinh khong ton tai trong do thi!" << endl;  return;  }  char newVertexName;  cout << "Nhap ten moi cho dinh " << oldVertexName << ": ";  cin >> newVertexName;  vertex[vertexIndex] = newVertexName;  cout << "Nhap thong tin moi cho dinh " << newVertexName << ": ";  for (int i = 0; i < n; i++) {  cin >> a[vertexIndex][i];  a[i][vertexIndex] = a[vertexIndex][i];  }  cout << "Thay doi thong tin va ten cua dinh " << oldVertexName << " thanh cong!" << endl;  cout << endl;  } |

Kế quả khi chạy chương trình:



Khi chọn số 4 và nhập các dữ liệu đầu vào:



Kết quả khi xuất ma trận:

