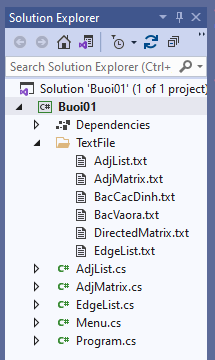
|  |  |
| --- | --- |
| Buổi 1 | **NHẬP, XUẤT & CÁC THAO TÁC CƠ BẢN TRÊN ĐỒ THỊ** |



**Tổ chức chương trình cho mỗi buổi thực hành**

* Mỗi buổi thực hành thực hiện trong một project, đặt tên : Buoi01, Buoi02, …
* Có tổ chức hệ thống menu, mỗi bài tập là một mục chọn
* Trong project có thư mục TextFile chứa các tập tin văn bản dạng .txt (là các input, output của chương trình)
* Các class được tạo trong thư mục chính project
* Class Menu.cs phục vụ tạo menu cho chương trình

**Bài 1. Bậc của các đỉnh trong đồ thị vô hướng**

Cho đồ thị vô hướng G = (V, E) có n (n ≤ 1000) đỉnh, các đỉnh được đánh số từ 1 đến n. Đồ thị G được lưu trong một file văn bản dưới dạng một ma trận kề. Hãy tổ chức cấu trúc dữ liệu ma trận kề để biểu diễn đồ thị G, và viết chương trình đọc đồ thị G từ file đã cho vào cấu trúc dữ liệu đó, sau đó tính bậc

của các đỉnh trong đồ thị G (bậc của một đỉnh là số cạnh liên thuộc với đỉnh đó).

|  |  |  |
| --- | --- | --- |
| Đồ thị | Input : AdjMatrix.txt | Output : BacCacdinh.txt |
|  | 6  0 1 0 0 0 1  1 0 1 0 0 1  0 1 0 1 1 1  0 0 1 0 1 0  0 0 1 1 0 1  1 1 1 0 1 0 | Bậc của các đỉnh  Đỉnh 0 : 2  1 : 3  2 : 4  3 : 2  4 : 3  5 : 4 |

Trong tất cả các bài tập, mỗi một kết quả ghi vào file output đều xuất lên màn hình

**Hướng dẫn**

1. Tạo project (console) Buoi01, trong project Buoi01 tạo thư mục TextFile.
2. Trong thư mục TextFile tạo file văn bản AdjMatrix.txt với dung như trên. Bước tạo project, thư mục TextFile và file .txt để làm input sẽ không nhắc lại từ các buổi sau (sinh viên tự làm)
3. Tạo class AdjMatrix.cs để tổ chức lưu trữ và xử lý các tính chất của đồ thị ma trận kề có hướng hoặc vô hướng với cấu trúc như sau :

AdjMatrix.cs

using System;

using System.IO;

using System.Text;

namespace Buoi01

{

class AdjMatrix

{

public int n; // số đỉmh

public int[,] a; // Ma trận kề

// propeties

public int N { get => n; set => n = value; }

public int[,] A { get => a; set => a = value; }

// constructor không đối số

public AdjMatrix() { }

// constructor có đối số k là số đỉnh của đồ thị

public AdjMatrix(int k)

{

n = k;

a = new int[n, n];

}

// Đọc file AdjMatrix --> ma trận a

public void FileToAdjMatrix(string filePath)

{

StreamReader sr = new StreamReader(filePath);

n = int.Parse(sr.ReadLine());

a = new int[n, n];

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

{

string[] s = sr.ReadLine().Split();

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

a[i, j] = int.Parse(s[j]);

}

sr.Close();

}

// Xuất ma trận a lên màn hình

public void Output()

{

Console.WriteLine("Đồ thị ma trận kề - số đỉnh : "+ n);

Console.WriteLine();

Console.Write(" Đỉnh |");

for (int i = 0; i < n; i++) Console.Write(" {0}", i);

Console.WriteLine(); Console.WriteLine(" " + new string('-',6\*n));

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

{

Console.Write(" {0} |", i);

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

Console.Write(" {0, 3}", a[i, j]);

Console.WriteLine();

}

}

// Các phương thức xử lý các thao tác trên đồ thị là bài tập thực hành

...

}

}

1. Trong class AdjMatrix.cs viết các phương thức xử lý thao tác tính bậc của các đỉnh

// Tính bậc của đỉnh i

public int DegVi(int i)

{

// Duyệt từng cột j trên dòng i

// Đếm số lượng ô(i, j) = 1

// Trả về kết quả

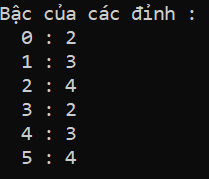
}

// Bậc của các đỉnh, tham số là tên file để ghi kết quả

public void DegVs(string filePath)

{

// Sử dụng đối tượng : StreamWriter sw = new StreamWriter(filePath);

 // Duyệt từng đỉnh của đồ thị

// Tính bậc của đỉnh i : DegVi(i);

// Ghi vào file filePath và xuất lên màn hình theo yêu cầu

// Đóng file

}

1. Thiết kế giao diện màn hình trong program.cs như sau :

using System;

using System.IO;

using System.Text;

namespace Buoi01

{

class Program

{

static void Main(string[] args)

{

// Xuất text theo Unicode (có dấu tiếng Việt)

Console.OutputEncoding = Encoding.Unicode;

// Nhập text theo Unicode (có dấu tiếng Việt)

Console.InputEncoding = Encoding.Unicode;

/\* Tạo menu \*/

Menu menu = new Menu();

string title = "NHẬP, XUẤT & CÁC THAO TÁC CƠ BẢN TRÊN ĐỒ THỊ"; // Tiêu đề menu

// Danh sách các mục chọn

string[] ms = { "1. Bài 1: Đồ thị vô hướng ma trận kề, tính bậc của của các đỉnh",

"2. Bài 2: Đồ thị có hướng ma trận kề, bậc vào, tính bậc ra của các đỉnh",

"3. Bài 3: Đồ thị danh sách kề, tính bậc các đỉnh",

"4. Bài 4: Đồ thị danh sách cạnh, tính bậc của các đỉnh",

"0. Thoát" };

int chon;

do

{

// Xuất menu

menu.ShowMenu(title, ms);

Console.Write(" Chọn : ");

chon = int.Parse(Console.ReadLine());

switch (chon)

{

case 1:

{ // Bài 1:

// Khai báo đối tượng : AdjMatrix g = new AdjMatrix();

// Tạo đường dẫn file input

string fileInput = "../../../TextFile/AdjMatrix.txt";

// Gọi các phương thức đọc file và xuất đồ thị

// Tạo đường dẫn file output

string fileOutput = "../../../TextFile/BacCacDinh.txt";

// Gọi phương thức tính bậc của các đỉnh và xuất

}

case 2:

{ // Bài 2:

}

case 3:

{ // Bài 3 :

}

case 4:

{ // Bài 4 :

}

}

Console.WriteLine(" Nhấn một phím bất kỳ");

Console.ReadKey();

Console.Clear();

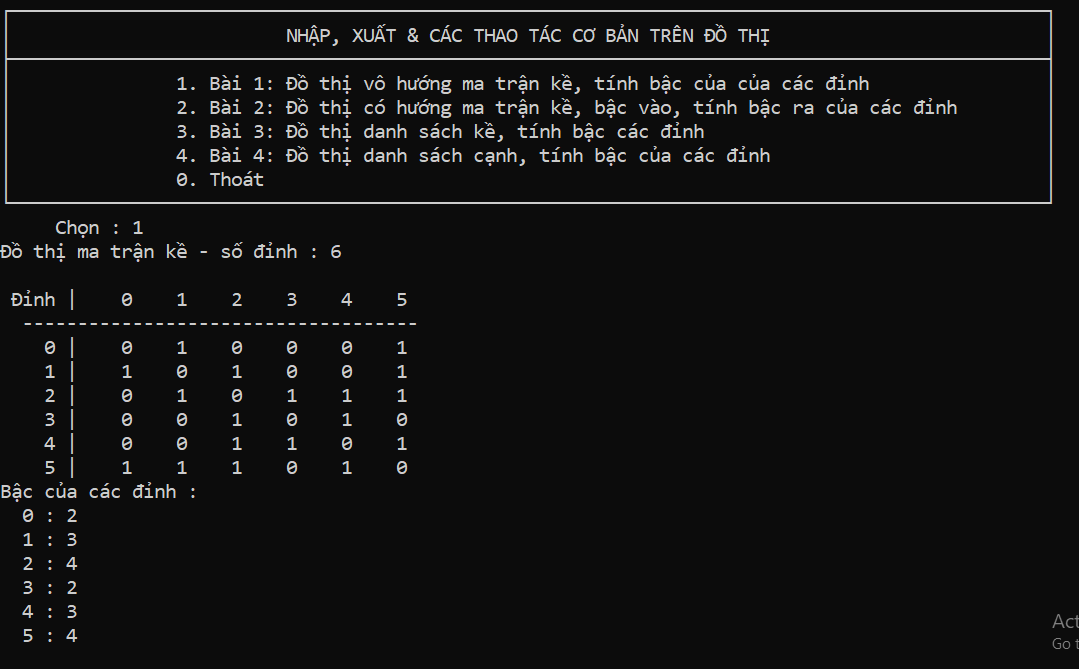
} while (chon != 0);

}

}

}

Kết quả thực thi với yêu cầu ghi file và màn hình như sau :



**Bài 2. Bậc vào, bậc ra**

Cho đồ thị có hướng G = (V, E) có n đỉnh được đánh số từ 1 đến n. Bậc vào của đỉnh x (x ∈ V) là số cung

đi vào đỉnh x. Bậc ra của đỉnh x là số cung đi ra đỉnh x. Hãy tính bậc vào và bậc ra của tất cả các đỉnh

trong đồ thị G.

|  |  |  |
| --- | --- | --- |
| Đồ thị | Input : DirectedMatrix.txt | Output (ghi file, xuất màn hình) |
|  | 6  0 1 1 0 0 0  0 0 0 1 1 0  0 1 0 0 1 0  0 0 0 0 1 1  0 1 0 0 0 1  0 0 0 0 0 0 | Bậc vào – bậc ra  0 : 0 - 2  1 : 3 - 2  2 : 1 - 2  3 : 1 - 2  4 : 3 - 2  5 : 2 - 0 |

**Hướng dẫn**

1. Trong class AdjMatrix.cs viết các phương thức :

// 1. Tính bậc ra của đỉnh i

public int DegOut(int i)

{

}

// 2. Tính bậc vào của đỉnh j

public int DegIn(int j)

{

}

// 3. Xuất bậc vào bậc ra của các đỉnh theo yêu cầu

public void DegInOut(string filePath)

{

}

1. Trong Program.cs, viết thêm phần thực thi bài 2 (case 2) với hình thức tương tự bài 1 (case 1)

**Bài 3. Danh sách kề**

Cho đồ thị vô hướng G = (V, E) có n (n ≤ 105) đỉnh, các đỉnh được đánh số từ 1 đến n và m cạnh (m ≤ 105). Hãy tổ chức cấu trúc dữ liệu cho đồ thị dưới dạng danh sách kề, và viết chương trình đọc đồ thị

G từ file đã cho, sau đó tính bậc của các đỉnh trong đồ thị G.

|  |  |  |
| --- | --- | --- |
| Đồ thị | Input : AdjList.txt | Output : Danhsachke.txt, xuất màn hình |
|  | 5  1 2  0 2 4  0 1 4  4  1 2 3 | Bậc của các đỉnh :  0 : 1  1 : 2  2 : 1  3 : 0  4 : 2 |

**Biểu diễn đồ thị có hướng và vô hướng bằng danh sách kề**

|  |  |  |
| --- | --- | --- |
| Đồ thị | AdjList.txt | AdjList : LinkedList<int>[] v; |
|  | 5  1 2  0 2 4  0 1 4  4  1 2 3 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Đỉnh | v |  |  |  |  |  |  | | 0 |  |  | 1 |  | 2 |  |  | | 1 |  |  | 0 |  | 2 |  | 4 | | 2 |  |  | 0 |  | 1 |  | 4 | | 3 |  |  | 4 |  |  |  |  | | 4 |  |  | 1 |  | 2 |  | 3 | |

* Đồ thị danh sách kề **v** là array với mỗi phần tử v[i] là một danh sách liên kết (dùng LinkedList của C#) mà mỗi phần tử là số nguyên <int> mô tả các đỉnh kề của đỉnh i (v[i])
* Với đỉnh i cô lập (không có đỉnh kề) 🡪 dòng i+1 trống và v[i] = null

**Hướng dẫn**

1. Trong TextFile tạo file DanhSachKe.txt với nội dung như trên
2. Trong Project, tạo class AdjList.cs như sau :

using System;

using System.IO;

using System.Text;

using System.Collections.Generic; // Thư viện cho đối tượng LinkedList

namespace Buoi01

{

class AdjList

{

LinkedList<int>[] v;

int n; // Số đỉnh

//Propeties

public int N { get => n; set => n = value; }

public LinkedList<int>[] V

{

get { return v; }

set { v = value; }

}

// Contructor

public AdjList() { }

public AdjList(int k) // Khởi tạo v có k đỉnh

{

v = new LinkedList<int>[k];

}

// copy g --> đồ thị hiện tại v

public AdjList(LinkedList<int>[] g)

{

v = g;

}

// Đọc file AdjList.txt --> danh sách kề v

public void FileToAdjList(string filePath)

{

StreamReader sr = new StreamReader(filePath);

n = int.Parse(sr.ReadLine());

v = new LinkedList<int>[n];

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

{

v[i] = new LinkedList<int>();

string st = sr.ReadLine();

// Đặt điều kiện không phải đỉnh cô lập

if (st != "")

{

string[] s = st.Split();

for (int j = 0; j < s.Length; j++)

{

int x = int.Parse(s[j]);

v[i].AddLast(x);

}

}

}

sr.Close();

}

// Xuất đồ thị

public void Output()

{

Console.WriteLine("Đồ thị danh sách kề - số đỉnh : " + n);

for (int i = 0; i < v.Length; i++)

{

Console.Write(" Đỉnh {0} ->", i);

foreach (int x in v[i])

Console.Write("{0, 3}", x);

Console.WriteLine();

}

}

// Phương thức tính bậc của các đỉnh, ghi file và xuất lên màn hình

}

}

1. Viết phương thức tính bậc của các đỉnh

public void DegV(string filePath)

{

// Khởi tạo biến sw là biến đối tượng StreamWriter

// Duyệt các đỉnh (các phần tử v[i])

// Bậc của đỉnh i = số phần tử của dslk v[i]

// Ghi file và xuất lên màn hình theo yêu cầu

// Đóng file

}

1. Trong Program.cs, viết thêm phần thực thi bài 3 (case 3) với hình thức tương tự bài 1 (case 1)

**Bài 4. Danh sách cạnh**

Cho đồ thị vô hướng G = (V, E) có n (n ≤ 105) đỉnh, các đỉnh được đánh số từ 1 đến n và m cạnh (m ≤ 105) được biểu diễn dưới dạng danh sách cạnh trong một file văn bản. Hãy tổ chức cấu trúc dữ liệu

cho đồ thị dưới dạng danh sách cạnh, và viết chương trình đọc đồ thị G từ file đã cho, sau đó tính bậc của các đỉnh trong đồ thị G.

**Biểu diễn đồ thị bằng danh sách cạnh**

|  |  |  |
| --- | --- | --- |
| Đồ thị | EdgeList.txt | EdgeList : LinkedList<Tuple<int, int>> g |
|  | **5** **6 (5 đỉnh,**  0 1 **6 cạnh**)  0 2  1 2  1 4  2 4  3 4 | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | g |  | (0,1) |  | (0,2) |  | (1,2) |  | (1,4) |  | (2,4) |  | (3,4) |   Đồ thị danh sách cạnh **g** là dslk mà mỗi phần tử là một bộ Tuple<int,int> là cặp số nguyên mô tả cạnh (i,j) |

Nếu là đồ thị có hướng 🡪 thứ tự (i,j) mô tả hướng của cạnh

**Hướng dẫn**

Class EdgeList.cs

using System;

using System.IO;

using System.Collections.Generic;

using System.Text;

namespace Buoi01

{

class EdgeList

{

LinkedList<Tuple<int, int>> g;

int n; // số đỉnh

int m; // số cạnh

// Propeties

public int N { get => n; set => n = value; }

public int M { get => m; set => m = value; }

public LinkedList<Tuple<int, int>> G { get => g; set => g = value; }

// constructor

public EdgeList()

{

g = new LinkedList<Tuple<int, int>>();

}

// Đọc file EdgeList.txt --> g

public void FileToEdgeList(string filePath)

{

StreamReader sr = new StreamReader(filePath);

string[] s = sr.ReadLine().Split();

n = int.Parse(s[0]);

m = int.Parse(s[1]);

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

{

s = sr.ReadLine().Split();

// khởi tạo một cạnh mới

Tuple<int, int> e = new Tuple<int, int>(int.Parse(s[0]), int.Parse(s[1]));

g.AddLast(e);

}

sr.Close();

}

// Xuất danh sách cạnh lên màn hình

public void Output()

{

Console.WriteLine("Danh sách cạnh của đồ thị với số đỉnh n = " + n );

foreach (Tuple<int, int> e in g)

Console.WriteLine(" (" + e.Item1 + "," + e.Item2 + ")");

}

// Tính bậc các đỉnh

public void DegV(string filePath)

{

}

}

}