**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC UEH**

**KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH**

🖎🕮✍



**ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI**

**ĐỒ THỊ VÀ ỨNG DỤNG TRONG QUẢN LÝ LỘ TRÌNH ĐƯỜNG ĐI**

**Học Phần: Cấu Trúc Dữ Liệu & Giải Thuật**

**Danh Sách Nhóm**:

1. NGUYỄN NGỌC BẢO
2. LỒ SÌN DẬU
3. ĐOÀN TRẦN BÁ ĐẠT
4. NGUYỄN HOÀN THIỆN

**Chuyên Ngành**: CÔNG NGHỆ PHẦN MỀM

**Khóa**: K46

**Giảng Viên**: TS. Đặng Ngọc Hoàng Thành

**Tp. Hồ Chí Minh, Ngày 19 tháng 12 năm 2021**

# MỤC LỤC

[MỤC LỤC 2](#_Toc90650895)

[CHƯƠNG 1. ĐỒ THỊ 3](#_Toc90650896)

[1.1. Các Khái Niệm Liên Quan 3](#_Toc90650897)

[1.2. Cấu Trúc và Cài Đặt Đồ Thị 3](#_Toc90650898)

[1.3. Các Thuật Toán Trên Đồ Thị 4](#_Toc90650899)

[a) Thuật Toán Tìm kiếm theo chiều sâu - Depth-First Search 4](#_Toc90650900)

[b) Thuật Toán Tìm kiếm theo chiều rộng - Breadth First Search 5](#_Toc90650901)

[c) Thuật Toán Tìm đường đi ngắn nhất 5](#_Toc90650902)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ LỚP 6](#_Toc90650903)

[2.1. Phân Tích Bài Toán Quản Lý Lộ Trình Đường Đi 6](#_Toc90650904)

[2.2. Sơ Đồ Lớp 7](#_Toc90650905)

[2.3. Cài Đặt Lớp 8](#_Toc90650906)

[CHƯƠNG 3. THIẾT KẾ GIAO DIỆN 20](#_Toc90650907)

[3.1. Giao Diện Menu Chính 20](#_Toc90650908)

[3.2. Chi Tiết Chức Năng 20](#_Toc90650909)

[CHƯƠNG 4. THẢO LUẬN & ĐÁNH GIÁ 24](#_Toc90650910)

[5.1. Các Kết Quả Nhận Được 24](#_Toc90650911)

[5.2. Một Số Tồn Tại 24](#_Toc90650912)

[5.1. Hướng Phát Triển 24](#_Toc90650913)

[PHỤ LỤC 25](#_Toc90650914)

[TÀI LIỆU THAM KHẢO 28](#_Toc90650915)

# CHƯƠNG 1. ĐỒ THỊ

## 1.1. Các Khái Niệm Liên Quan

* Đồ thị (Graph) là một dạng cấu trúc dữ liệu được cấu tạo từ tập các đỉnh V (vertex) và tập các cạnh E (edge): G=(V, E).
* Hai đỉnh được nối với nhau tạo thành một cặp đỉnh (pair). Mỗi cặp đỉnh như vậy tạo thành một cạnh. Một cạnh được tạo từ một đỉnh duy nhất gọi là khuyên. Nếu tồn tại hai cạnh trên cùng một cặp điểm, thì cặp cạnh đó gọi là cặp cạnh song song (hoặc cạnh bội).
* Một đồ thì mà trong đó các cặp đỉnh là các bộ sắp thứ tự được gọi Đồ thị có hướng (Digraph). Ngược lại, được gọi là Đồ thị vô hướng.
* Nếu mỗi cạnh của đồ thị được gán với một trọng số (weight), thì đồ thị đó được gọi là Đồ thị có trọng số (hoặc là mạng lưới network).
* Đơn đồ thị là đồ thị (có hướng hoặc vô hướng) KHÔNG chứa khuyên hoặc cạnh bội.
* Để biểu diễn đồ thị: biển diễn hình học, biểu diễn bằng ma trận liền kề.

## 1.2. Cấu Trúc và Cài Đặt Đồ Thị

|  |  |
| --- | --- |
| **Lớp Đỉnh Vertex**  public class **Vertex**  {  public bool wasVisited;  public string label;  public Vertex(string label)  {  this.label = label;  wasVisited = false;  }  } | |
| public class **Graph**  {  private const int NUM\_VERTICES = 20;  private Vertex[] vertices;  private int[,] adjMatrix;  int numVerts;  public Graph()  {  vertices = new **Vertex**[NUM\_VERTICES];  adjMatrix = new int[NUM\_VERTICES, NUM\_VERTICES];  numVerts = 0;  for (int j = 0; j < NUM\_VERTICES; j++)  for (int k = 0; k < NUMVERTICES; k++)  adjMatrix[j, k] = 0;  } | public void AddVertex(string label)  {  vertices[numVerts] = new **Vertex**(label);  numVerts++;  }  public void AddEdge(int start, int eend)  {  adjMatrix[start, eend] = 1;  adjMatrix[eend, start] = 1;  }  public void ShowVertex(int v)  {  Console.Write(vertices[v].label + " ");  }  } |

## 1.3. Các Thuật Toán Trên Đồ Thị

### a) Thuật Toán Tìm kiếm theo chiều sâu - Depth-First Search

|  |  |
| --- | --- |
| PSEUDOCODE | |
| procedure GetAdjUnvisitedVertex(int v):  FOR j = 0 to (numVertices – 1)  IF adjMatrix(v,j) == 1 &&  vertices[j].WasVisited == false THEN  return j  ENDIF  ENDFOR  return -1  procedure DepthFirstSearch():  vertices[0].WasVisited = true  ShowVertex(0)  Stack gStack = new Stack()  gStack.Push(0) | WHILE gStack.Count > 0  v = GetAdjUnvisitedVertex(gStack.Peek())  IF v == -1 THEN  gStack.Pop()  ELSE  vertices[v].WasVisited = true  ShowVertex(v)  gStack.Push(v)  ENDIF  FOR j = 0 to (numVertices – 1)  vertices[j].WasVisited = false  ENDFOR |

### b) Thuật Toán Tìm kiếm theo chiều rộng - Breadth First Search

|  |  |
| --- | --- |
| PSEUDOCODE | |
| procedure BreadthFirstSearch():  gQueue = new Queue()  vertices[0].WasVisited = true  ShowVertex(0)  gQueue.EnQueue(0)  WHILE gQueue.Count > 0  vert1 = gQueue.Dequeue()  vert2 = GetAdjUnvisitedVertex(vert1) | WHILE vert2! = -1  vertices[vert2].WasVisited = true  ShowVertex(vert2)  gQueue.Enqueue(vert2)  vert2 = GetAdjUnvisitedVertex(vert1)  ENDWHILE  ENDWHILE  FOR i = 0 to (numVertices – 1)  vertices[index].WasVisited = false  ENDFOR |

### c) Thuật Toán Tìm đường đi ngắn nhất

|  |  |
| --- | --- |
| PSEUDOCODE | |
| procedure Path():  startTree = 0  vertexList[startTree].isInTree = true  nTree = 1  FOR j = 0 to nVerts  tempDist = adjMat[startTree, j]  sPath[j] = new DistOriginal(startTree, tempDist)  ENDFOR  WHILE nTree < nVerts  indexMin = GetMin() | minDist = sPath[indexMin].distance  currentVert = indexMin  startToCurrent = sPath[indexMin].distance  vertexList[currentVert].isInTree = true  nTree++  AdjustShortPath()  ENDWHILE  DisplayPaths()  nTree = 0  FOR j = 0 to (nVerts – 1)  vertexList[j].isInTree = false  ENDFOR |

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ LỚP

## 2.1. Phân Tích Bài Toán Quản Lý Lộ Trình Đường Đi

## 2.2. Sơ Đồ Lớp

Diagram

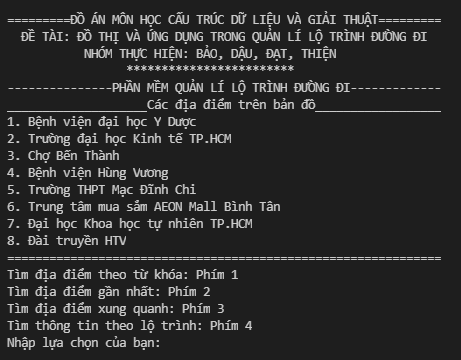
Description automatically generated

## 2.3. Cài Đặt Lớp

|  |  |
| --- | --- |
| Class Vertex | Class DistOriginal |
| public class Vertex  {      public Map label;      public bool isInTree;      public Vertex(Map lab)      {          label = lab;          isInTree = false;      }  } | public class DistOriginal  {      public int distance;      public int parentVert;      public DistOriginal(int pv, int d)      {          distance = d;          parentVert = pv;      }  } |
| Class Car | Class Motorbike |
| namespace baithi  {      public class Car      {          private string id;          private string name;          private float speed;          private float price;          public string getId()          {              return id;          }          public string getName()          {              return name;          }          public float getSpeed()          {              return speed;          }          public float getPrice()          {              return price;          }          public Car(string id, string name, float speed, float price)          {              this.id = id;              this.name = name;              this.speed = speed;              this.price = price;          }      }  } | namespace baithi  {      public class Motorbike      {          private string id;          private string name;          private float speed;          private float price;          public string getId()          {              return id;          }          public string getName()          {              return name;          }          public float getSpeed()          {              return speed;          }          public float getPrice()          {              return price;          }          public Motorbike(string id, string name, float speed, float price)          {              this.id = id;              this.name = name;              this.speed = speed;              this.price = price;          }      }  } |
| Class Map | |
| public class Map  {      private float x;      private float y;      private string name;      private int id;      private string city;      private string district;      public float getX()      {          return x;      }      public float getY()      {          return y;      }      public string getName()      {          return name;      }      public int getId()      {          return id;      }      public string getCity()      {          return city;      }      public string getDistrict()      {          return district;      }      public Map(float x, float y, string name, int id, string district, string city)      {          this.x = x;          this.y = y;          this.name = name;          this.id = id;          this.city = city;          this.district = district;      }      public override string ToString()      {          return name + " ((" + x + "; " + y + "), " + district + ", " + city + ")";      }  } | |
| Class Graph | |
| using System;  using System.Collections.Generic;  namespace baithi  {      public class Program      {          static void ChooseEnd(Graph map)          {              System.Console.Write("Chọn địa điểm cần đến: ");              int end = Int32.Parse(Console.ReadLine());              if (end >= 1 && end <= 7)              {                  map.DisplayMenu(end);              }              else              {                  Console.WriteLine("\nYêu cầu bạn nhập không đúng. Chỉ nhập từ 1 - 7");                  ChooseEnd(map);              }          }          static void Main(string[] args)          {              Console.Clear();              Console.OutputEncoding = System.Text.Encoding.Unicode;              Console.InputEncoding = System.Text.Encoding.Unicode;              Map bv\_daihoc = new Map(50, 20, "Bệnh viện Đại học Y dược", 0, "Quận 5", "TP.HCM"); //0              Map ueh = new Map(40, 53.2f, "Đại học kinh tế TP.HCM CSB", 1, "Quận 10", "TP.HCM"); //1              Map chobenthanh = new Map(82.4f, 20, "Chợ Bến Thành", 2, "Quận 1", "TP.HCM"); //2              Map bv\_hungvuong = new Map(50, 20, "Bệnh viện Hùng Vương", 3, "Quận 5", "TP.HCM"); //3              Map macdinhchi = new Map(20, 30.5f, "Trường THPT Mạc Đĩnh Chi", 4, "Quận 6", "TP.HCM"); //4              Map aeon = new Map(20.3f, 302.4f, "Trung tâm mua sắm AEON Mall Bình Tân", 5, "Bình Tân", "TP.HCM"); //5              Map khtn = new Map(49.2f, 90.2f, "Đại học Khoa học tự nhiên TPHCM", 6, "Quận 8", "TP.HCM"); //6              Map htv = new Map(20.4f, 30.2f, "Đài truyền hình HTV", 7, "Quận 1", "TP.HCM"); //7              List<Map> listpos = new List<Map>();              listpos.Add(bv\_daihoc); listpos.Add(ueh);              listpos.Add(chobenthanh); listpos.Add(bv\_hungvuong);              listpos.Add(macdinhchi); listpos.Add(aeon);              listpos.Add(khtn); listpos.Add(htv);              Graph mapnear = new Graph();              mapnear.AddVertex(bv\_daihoc);              mapnear.AddVertex(ueh);              mapnear.AddVertex(chobenthanh);              mapnear.AddVertex(bv\_hungvuong);              mapnear.AddVertex(macdinhchi);              mapnear.AddVertex(aeon);              mapnear.AddVertex(khtn);              mapnear.AddVertex(htv);              mapnear.AddEdge(0, 1, 2); mapnear.AddEdge(0, 3, 4); mapnear.AddEdge(0, 4, 2);              mapnear.AddEdge(1, 2, 6); mapnear.AddEdge(1, 4, 3); mapnear.AddEdge(1, 6, 4); mapnear.AddEdge(1, 0, 2);              mapnear.AddEdge(2, 5, 12); mapnear.AddEdge(2, 6, 7); mapnear.AddEdge(2, 7, 5); mapnear.AddEdge(2, 1, 6);              mapnear.AddEdge(3, 5, 3); mapnear.AddEdge(3, 6, 9); mapnear.AddEdge(3, 0, 4);              mapnear.AddEdge(4, 5, 1); mapnear.AddEdge(4, 7, 14); mapnear.AddEdge(4, 1, 3); mapnear.AddEdge(4, 0, 2);              mapnear.AddEdge(5, 6, 8); mapnear.AddEdge(5, 2, 12); mapnear.AddEdge(5, 3, 3); mapnear.AddEdge(5, 4, 1);              mapnear.AddEdge(6, 7, 4); mapnear.AddEdge(6, 1, 4); mapnear.AddEdge(6, 2, 7); mapnear.AddEdge(6, 3, 9); mapnear.AddEdge(6, 5, 8);              mapnear.AddEdge(7, 2, 5); mapnear.AddEdge(7, 4, 14); mapnear.AddEdge(7, 6, 4);          Begin:              Console.Clear();              System.Console.WriteLine("=========ĐỒ ÁN MÔN HỌC CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT=========");              System.Console.WriteLine("  ĐỀ TÀI: ĐỒ THỊ VÀ ỨNG DỤNG TRONG QUẢN LÍ LỘ TRÌNH ĐƯỜNG ĐI  ");              System.Console.WriteLine("           NHÓM THỰC HIỆN: BẢO, DẬU, ĐẠT, THIỆN");              System.Console.WriteLine("                 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*                     ");              System.Console.WriteLine("---------------PHẦN MỀM QUẢN LÍ LỘ TRÌNH ĐƯỜNG ĐI-------------");              System.Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Các địa điểm trên bản đồ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");              System.Console.WriteLine("1. Bệnh viện đại học Y Dược");              System.Console.WriteLine("2. Trường đại học Kinh tế TP.HCM");              System.Console.WriteLine("3. Chợ Bến Thành");              System.Console.WriteLine("4. Bệnh viện Hùng Vương");              System.Console.WriteLine("5. Trường THPT Mạc Đĩnh Chi");              System.Console.WriteLine("6. Trung tâm mua sắm AEON Mall Bình Tân");              System.Console.WriteLine("7. Đại học Khoa học tự nhiên TP.HCM");              System.Console.WriteLine("8. Đài truyền HTV");              System.Console.WriteLine("==============================================================");              System.Console.WriteLine("Tìm địa điểm theo từ khóa: Phím 1");              System.Console.WriteLine("Tìm địa điểm gần nhất: Phím 2");              System.Console.WriteLine("Tìm địa điểm xung quanh: Phím 3");              System.Console.WriteLine("Tìm thông tin theo lộ trình: Phím 4");          Error1:              System.Console.Write("Nhập lựa chọn của bạn: ");              int choose = int.Parse(Console.ReadLine());              switch (choose)              {                  case 1:                      Console.Clear();                      Console.Write("Nhập từ khóa cần tìm kiếm: ");                      string text = Console.ReadLine();                      System.Console.WriteLine("------------------------------------");                      Console.WriteLine("Địa điểm bạn cần tìm là: ");                      for (int i = 0; i < listpos.Count; i++)                      {                          if (listpos[i].getName().ToLower().Contains(text.ToLower()))                          {                              Console.WriteLine(listpos[i].getName());                          }                      }                      break;                  case 2:                      Console.Clear();                      mapnear.PrintDes(-1);                  FError1:                      System.Console.Write("Nhập địa điểm bạn muốn: ");                      int pos1 = Int32.Parse(Console.ReadLine()) - 1;                      System.Console.WriteLine("------------------------------------");                      if (pos1 >= 0 && pos1 <= 7)                      {                          mapnear.DisplayNearestPos(pos1);                      }                      else                      {                          System.Console.WriteLine("\nYêu cầu bạn nhập không đúng vui lòng nhập lại! (1 - 8)");                          goto FError1;                      }                      break;                  case 3:                      Console.Clear();                      mapnear.PrintDes(-1);                  FError2:                      System.Console.Write("Nhập địa điểm bạn muốn: ");                      int pos2 = Int32.Parse(Console.ReadLine()) - 1;                      System.Console.WriteLine("------------------------------------");                      if (pos2 >= 0 && pos2 <= 7)                      {                          mapnear.DisplayNearPos(pos2);                      }                      else                      {                          System.Console.WriteLine("\nYêu cầu bạn nhập không đúng vui lòng nhập lại! (1 - 7)");                          goto FError2;                      }                      break;                  case 4:                      Console.Clear();                      System.Console.WriteLine("-----------Các địa điểm----------");                      System.Console.WriteLine("Bệnh viện đại học Y Dược: 0");                      System.Console.WriteLine("Đại học kinh tế TP.HCM: 1");                      System.Console.WriteLine("Chợ Bến Thành: 2");                      System.Console.WriteLine("Bệnh viện Hùng Vương: 3");                      System.Console.WriteLine("Trường THPT Mạc Đĩnh Chi: 4");                      System.Console.WriteLine("Trung tâm mua sắm AEON Mall Bình Tân: 5");                      System.Console.WriteLine("Đại học Khoa học tự nhiên TPHCM: 6");                      System.Console.WriteLine("Đài truyền hình HTV: 7");                  FError3:                      System.Console.Write("Chọn địa điểm xuất phát: ");                      int start = Int32.Parse(Console.ReadLine());                      switch (start)                      {                          case 0:                              Console.Clear();                              System.Console.WriteLine("Điểm xuất phát: " + bv\_daihoc.getName());                              mapnear.PrintDes(start);                              ChooseEnd(mapnear);                              break;                          case 1:                              Console.Clear();                              Graph map1 = new Graph();                              map1.AddVertex(ueh);                              map1.AddVertex(bv\_daihoc);                              map1.AddVertex(chobenthanh);                              map1.AddVertex(bv\_hungvuong);                              map1.AddVertex(macdinhchi);                              map1.AddVertex(aeon);                              map1.AddVertex(khtn);                              map1.AddVertex(htv);                              map1.AddEdge(0, 1, 2); map1.AddEdge(0, 2, 6); map1.AddEdge(0, 4, 3); map1.AddEdge(0, 6, 4);                              map1.AddEdge(1, 3, 4); map1.AddEdge(1, 4, 2); map1.AddEdge(1, 0, 2);                              map1.AddEdge(2, 5, 12); map1.AddEdge(2, 6, 7); map1.AddEdge(2, 7, 5); map1.AddEdge(2, 0, 6);                              map1.AddEdge(3, 5, 3); map1.AddEdge(3, 6, 9); map1.AddEdge(3, 1, 4);                              map1.AddEdge(4, 5, 1); map1.AddEdge(4, 7, 14); map1.AddEdge(4, 1, 2); map1.AddEdge(4, 0, 3);                              map1.AddEdge(5, 6, 8); map1.AddEdge(5, 2, 12); map1.AddEdge(5, 3, 3); map1.AddEdge(5, 4, 1);                              map1.AddEdge(6, 7, 4); map1.AddEdge(6, 2, 7); map1.AddEdge(6, 3, 9); map1.AddEdge(6, 5, 8); map1.AddEdge(6, 0, 4);                              map1.AddEdge(7, 2, 5); map1.AddEdge(7, 4, 14); map1.AddEdge(7, 6, 4);                              System.Console.WriteLine("Điểm xuất phát: " + ueh.getName());                              map1.PrintDes(start);                              ChooseEnd(map1);                              break;                          case 2:                              Console.Clear();                              Graph map2 = new Graph();                              map2.AddVertex(chobenthanh); map2.AddVertex(ueh);                              map2.AddVertex(bv\_daihoc); map2.AddVertex(bv\_hungvuong);                              map2.AddVertex(macdinhchi); map2.AddVertex(aeon);                              map2.AddVertex(khtn); map2.AddVertex(htv);                              map2.AddEdge(0, 1, 6); map2.AddEdge(0, 5, 12); map2.AddEdge(0, 6, 7); map2.AddEdge(0, 7, 5);                              map2.AddEdge(1, 2, 2); map2.AddEdge(1, 4, 3); map2.AddEdge(1, 6, 4); map2.AddEdge(1, 0, 6);                              map2.AddEdge(2, 3, 4); map2.AddEdge(2, 4, 2); map2.AddEdge(2, 1, 2);                              map2.AddEdge(3, 5, 3); map2.AddEdge(3, 6, 9); map2.AddEdge(3, 2, 4);                              map2.AddEdge(4, 5, 1); map2.AddEdge(4, 7, 14); map2.AddEdge(4, 1, 3); map2.AddEdge(4, 2, 2);                              map2.AddEdge(5, 6, 8); map2.AddEdge(5, 3, 3); map2.AddEdge(5, 4, 1); map2.AddEdge(5, 0, 12);                              map2.AddEdge(6, 7, 4); map2.AddEdge(6, 1, 4); map2.AddEdge(6, 3, 9); map2.AddEdge(6, 5, 8); map2.AddEdge(6, 0, 7);                              map2.AddEdge(7, 4, 14); map2.AddEdge(7, 6, 4); map2.AddEdge(7, 0, 5);                              System.Console.WriteLine("Điểm xuất phát: " + chobenthanh.getName());                              map2.PrintDes(start);                              ChooseEnd(map2);                              break;                          case 3:                              Console.Clear();                              Graph map3 = new Graph();                              map3.AddVertex(bv\_hungvuong); map3.AddVertex(bv\_daihoc);                              map3.AddVertex(ueh); map3.AddVertex(chobenthanh);                              map3.AddVertex(macdinhchi); map3.AddVertex(aeon);                              map3.AddVertex(khtn); map3.AddVertex(htv);                              map3.AddEdge(0, 1, 4); map3.AddEdge(0, 5, 3); map3.AddEdge(0, 6, 9);                              map3.AddEdge(1, 4, 2); map3.AddEdge(1, 2, 2); map3.AddEdge(1, 0, 4);                              map3.AddEdge(2, 4, 3); map3.AddEdge(2, 6, 4); map3.AddEdge(2, 3, 6); map3.AddEdge(2, 1, 2);                              map3.AddEdge(3, 5, 12); map3.AddEdge(3, 6, 7); map3.AddEdge(3, 7, 4); map3.AddEdge(3, 2, 6);                              map3.AddEdge(4, 5, 1); map3.AddEdge(4, 7, 14); map3.AddEdge(4, 1, 2); map3.AddEdge(4, 2, 3);                              map3.AddEdge(5, 6, 8); map3.AddEdge(5, 3, 12); map3.AddEdge(5, 4, 1); map3.AddEdge(5, 0, 3);                              map3.AddEdge(6, 7, 4); map3.AddEdge(6, 2, 4); map3.AddEdge(6, 3, 7); map3.AddEdge(6, 5, 8); map3.AddEdge(6, 0, 9);                              map3.AddEdge(7, 3, 4); map3.AddEdge(7, 4, 14); map3.AddEdge(7, 6, 4);                              System.Console.WriteLine("Điểm xuất phát: " + bv\_hungvuong.getName());                              map3.PrintDes(start);                              ChooseEnd(map3);                              break;                          case 4:                              Console.Clear();                              Graph map4 = new Graph();                              map4.AddVertex(macdinhchi); map4.AddVertex(bv\_daihoc);                              map4.AddVertex(ueh); map4.AddVertex(chobenthanh);                              map4.AddVertex(bv\_hungvuong); map4.AddVertex(aeon);                              map4.AddVertex(khtn); map4.AddVertex(htv);                              map4.AddEdge(0, 1, 2); map4.AddEdge(0, 2, 3); map4.AddEdge(0, 5, 1); map4.AddEdge(0, 7, 14);                              map4.AddEdge(1, 4, 4); map4.AddEdge(1, 2, 2); map4.AddEdge(1, 0, 2);                              map4.AddEdge(2, 6, 4); map4.AddEdge(2, 3, 6); map4.AddEdge(2, 1, 2); map4.AddEdge(2, 0, 3);                              map4.AddEdge(3, 5, 12); map4.AddEdge(3, 6, 7); map4.AddEdge(3, 7, 5); map4.AddEdge(3, 2, 6);                              map4.AddEdge(4, 5, 3); map4.AddEdge(4, 6, 9); map4.AddEdge(4, 1, 4);                              map4.AddEdge(5, 6, 8); map4.AddEdge(5, 3, 12); map4.AddEdge(5, 4, 3); map4.AddEdge(5, 0, 1);                              map4.AddEdge(6, 7, 4); map4.AddEdge(6, 2, 4); map4.AddEdge(6, 3, 7); map4.AddEdge(6, 4, 9); map4.AddEdge(6, 5, 8);                              map4.AddEdge(7, 3, 5); map4.AddEdge(7, 6, 4); map4.AddEdge(7, 0, 14);                              System.Console.WriteLine("Điểm xuất phát: " + macdinhchi.getName());                              map4.PrintDes(start);                              ChooseEnd(map4);                              break;                          case 5:                              Console.Clear();                              Graph map5 = new Graph();                              map5.AddVertex(aeon); map5.AddVertex(bv\_hungvuong);                              map5.AddVertex(bv\_daihoc); map5.AddVertex(ueh);                              map5.AddVertex(macdinhchi); map5.AddVertex(chobenthanh);                              map5.AddVertex(khtn); map5.AddVertex(htv);                              map5.AddEdge(0, 1, 3); map5.AddEdge(0, 4, 1); map5.AddEdge(0, 6, 8); map5.AddEdge(0, 5, 2);                              map5.AddEdge(1, 2, 4); map5.AddEdge(1, 6, 9); map5.AddEdge(1, 0, 3);                              map5.AddEdge(2, 4, 2); map5.AddEdge(2, 3, 2); map5.AddEdge(2, 1, 4);                              map5.AddEdge(3, 4, 3); map5.AddEdge(3, 6, 4); map5.AddEdge(3, 5, 6); map5.AddEdge(3, 2, 2);                              map5.AddEdge(4, 7, 14); map5.AddEdge(4, 2, 2); map5.AddEdge(4, 3, 3); map5.AddEdge(4, 0, 1);                              map5.AddEdge(5, 6, 7); map5.AddEdge(5, 7, 5); map5.AddEdge(5, 3, 6); map5.AddEdge(5, 0, 2);                              map5.AddEdge(6, 7, 4); map5.AddEdge(6, 1, 9); map5.AddEdge(6, 3, 4); map5.AddEdge(6, 5, 7); map5.AddEdge(6, 0, 8);                              map5.AddEdge(7, 4, 14); map5.AddEdge(7, 5, 5); map5.AddEdge(7, 6, 4);                              System.Console.WriteLine("Điểm xuất phát: " + aeon.getName());                              map5.PrintDes(start);                              ChooseEnd(map5);                              break;                          case 6:                              Console.Clear();                              Graph map6 = new Graph();                              map6.AddVertex(khtn); map6.AddVertex(htv);                              map6.AddVertex(chobenthanh); map6.AddVertex(ueh);                              map6.AddVertex(macdinhchi); map6.AddVertex(aeon);                              map6.AddVertex(bv\_hungvuong); map6.AddVertex(bv\_daihoc);                              map6.AddEdge(0, 1, 4); map6.AddEdge(0, 2, 7); map6.AddEdge(0, 3, 4); map6.AddEdge(0, 5, 8); map6.AddEdge(0, 6, 9);                              map6.AddEdge(1, 4, 14); map6.AddEdge(1, 2, 5); map6.AddEdge(1, 0, 4);                              map6.AddEdge(2, 3, 6); map6.AddEdge(2, 5, 12); map6.AddEdge(2, 1, 5); map6.AddEdge(2, 0, 7);                              map6.AddEdge(3, 4, 3); map6.AddEdge(3, 7, 2); map6.AddEdge(3, 2, 6); map6.AddEdge(3, 0, 4);                              map6.AddEdge(4, 7, 2); map6.AddEdge(4, 5, 1); map6.AddEdge(4, 1, 14); map6.AddEdge(4, 3, 3);                              map6.AddEdge(5, 6, 3); map6.AddEdge(5, 2, 12); map6.AddEdge(5, 4, 1); map6.AddEdge(5, 0, 8);                              map6.AddEdge(6, 7, 4); map6.AddEdge(6, 5, 3); map6.AddEdge(6, 0, 9);                              map6.AddEdge(7, 3, 2); map6.AddEdge(7, 4, 2); map6.AddEdge(7, 6, 4);                              System.Console.WriteLine("Điểm xuất phát: " + khtn.getName());                              map6.PrintDes(start);                              ChooseEnd(map6);                              break;                          case 7:                              Console.Clear();                              Graph map7 = new Graph();                              map7.AddVertex(htv); map7.AddVertex(khtn);                              map7.AddVertex(chobenthanh); map7.AddVertex(ueh);                              map7.AddVertex(macdinhchi); map7.AddVertex(aeon);                              map7.AddVertex(bv\_hungvuong); map7.AddVertex(bv\_daihoc);                              map7.AddEdge(0, 1, 4); map7.AddEdge(0, 4, 14); map7.AddEdge(0, 2, 5);                              map7.AddEdge(1, 2, 7); map7.AddEdge(1, 3, 4); map7.AddEdge(1, 5, 8); map7.AddEdge(1, 6, 9); map7.AddEdge(1, 0, 4);                              map7.AddEdge(2, 5, 12); map7.AddEdge(2, 3, 6); map7.AddEdge(2, 1, 7); map7.AddEdge(2, 0, 5);                              map7.AddEdge(3, 4, 3); map7.AddEdge(3, 7, 2); map7.AddEdge(3, 1, 4); map7.AddEdge(3, 2, 6);                              map7.AddEdge(4, 7, 2); map7.AddEdge(4, 5, 1); map7.AddEdge(4, 3, 3); map7.AddEdge(4, 0, 14);                              map7.AddEdge(5, 6, 3); map7.AddEdge(5, 1, 8); map7.AddEdge(5, 2, 12); map7.AddEdge(5, 4, 1);                              map7.AddEdge(6, 7, 4); map7.AddEdge(6, 1, 9); map7.AddEdge(6, 5, 3);                              map7.AddEdge(7, 3, 2); map7.AddEdge(7, 4, 2); map7.AddEdge(7, 6, 4);                              System.Console.WriteLine("Điểm xuất phát: " + htv.getName());                              map7.PrintDes(start);                              ChooseEnd(map7);                              break;                          default:                              System.Console.WriteLine("\nYêu cầu bạn nhập không đúng vui lòng nhập lại! (1 - 7)");                              goto FError3;                      }                      break;                  default:                      System.Console.WriteLine("------------------------------------");                      System.Console.WriteLine("Yêu cầu bạn nhập không đúng vui lòng nhập lại! (1 - 4)");                      goto Error1;              }          Error2:              System.Console.Write("Để tiếp tục sử dụng chương trình ấn phím 1, để thoát chương trình ấn phím 2: ");              int end = Int32.Parse(Console.ReadLine());              switch (end)              {                  case 1:                      goto Begin;                  case 2:                      Console.Clear();                      System.Console.WriteLine("Bạn đã thoát chương trình");                      break;                  default:                      System.Console.WriteLine("------------------------------------");                      System.Console.WriteLine("Yêu cầu bạn nhập không đúng vui lòng nhập lại! (1 - 2)");                      goto Error2;              }          }      }  } | |

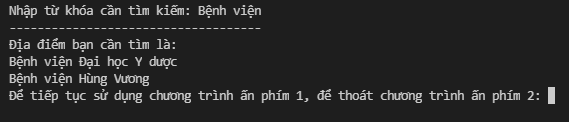
# CHƯƠNG 3. THIẾT KẾ GIAO DIỆN

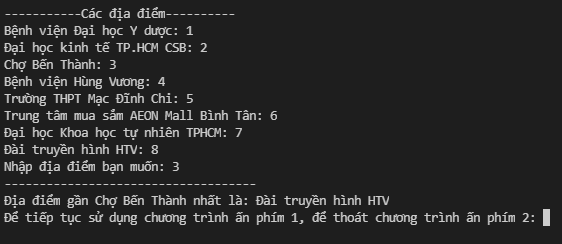
## 3.1. Giao Diện Menu Chính



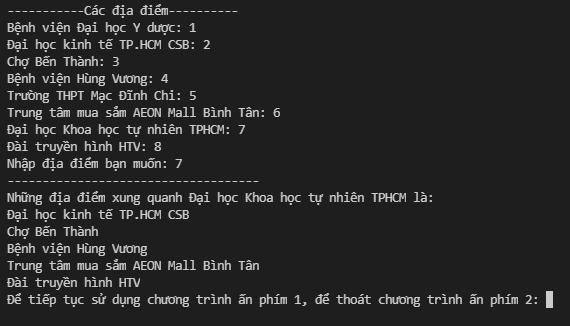
## 3.2. Chi Tiết Chức Năng

1. Tìm địa điểm theo từ khóa



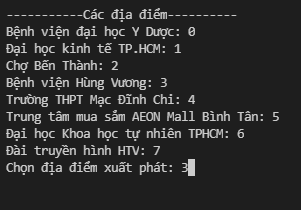
2. Tìm địa điểm gần nhất

3. Tìm địa điểm xung quanh

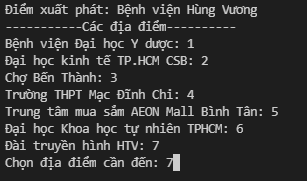


4. Tìm thông tin theo lộ trình

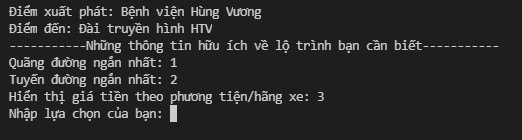
Chọn địa điểm xuất phát



Chọn điểm đến:



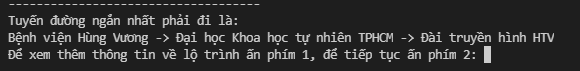
Các chức năng:



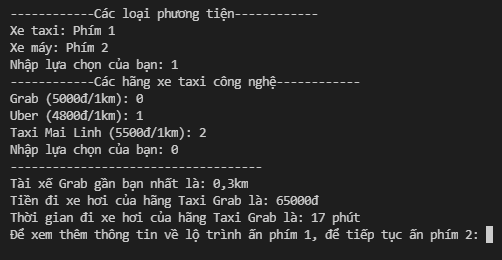
a. Quãng đường ngắn nhất



b. Tuyến đường ngắn nhất



c. Hiển thị giá theo phương tiện/ hãng xe



# CHƯƠNG 4. THẢO LUẬN & ĐÁNH GIÁ

## 4.1. Các Kết Quả Nhận Được

- Người dùng có thể tự nhập địa điểm cần tìm hoặc có thể tìm kiếm theo một vài từ khóa đã được gợi ý sẵn.

- Người dùng có thể biết được địa điểm gần nhất với vị trí mặc định mà người dùng đã chọn.

- Đồng thời chương trình cũng có thể cung cấp thông tin những địa điểm xung quanh địa điểm mặc định

- Cuối cùng thì người dùng có thể chọn điểm bắt đầu và điểm kết thúc cũng như sử dụng những tính năng đi kèm như chọn phương tiên vận chuyển…

-Người dùng có thể biết được đường đi ngắn nhất đi từ điểm xuất phát đến điểm kết thúc và biết được các tuyến đường cần phải đi

## 4.2. Một Số Tồn Tại

-Không thể nhập quá nhiều địa điểm, bởi việc nhập nhiều sẽ khiến dữ liệu bị nhiều, dẫn đến việc tốn nhiều dòng code cũng như dung lượng hơn để thực hiện các chức năng.

## 4.3. Hướng Phát Triển

- Chúng ta có thể mở rộng thêm nhiều địa điểm hơn giữa các tỉnh thành hoặc giữa các quốc gia trong tương lai. Đồng thời cũng có thể phát triển thêm chức năng tìm đường đi ngắn thứ hai.

# PHỤ LỤC

- Đưa Link Toàn Bộ Mã Nguồn Lên GitHub và Đưa Link Vào Đây

[bao201102/Doan\_CTDL (github.com)](https://github.com/bao201102/Doan_CTDL)

- Viết Hướng Dẫn Cách Cài Đặt Để Chạy

Bước 1: Tải git về máy tính [tại đây](https://git-scm.com/downloads) (chọn theo hệ điều hành phù hợp) và cài đặt.

Bước 2:

* ***Đối với Visual Studio Code***

Tại Terminal gõ lệnh **git clone** [**https://github.com/bao201102/Doan\_CTDL.git**](https://github.com/bao201102/Doan_CTDL.git)Text

Description automatically generated

(Lấy link theo như hình).

A screenshot of a computer

Description automatically generated with medium confidence

* ***Đối với Visual Studio***

Nhấp vào **Clone a repository** sau đó lấy link như trên

A screenshot of a computer

Description automatically generated with medium confidence

Sau đó lấy đường link như trên và dán link vào **Repository location** rồi ấn Clone.Graphical user interface, text, application

Description automatically generated

Bước 3: Sau khi đã clone source code thì bắt đầu chạy chương trình

- Phân Công Công Việc

|  |  |
| --- | --- |
| **Thành Viên** | **Nhiệm Vụ** |
| Nguyễn Ngọc Bảo | PrintPath  Giao diện (program.cs)  Thiết kế Class Diagram  Chương 2 bản báo cáo |
| Lồ Sìn Dậu | DisplayNearestPos  DisplayNearPos  Tối ưu code  Chương 1 bản báo cáo |
| Đoàn Trần Bá Đạt | DisplayCost (Class Car, Motorbike, Map)  Cài đặt cấu trúc đồ thị  Tùy chọn lộ trình (điểm xuất phát, điểm kết thúc)  Đường đi ngắn nhất  Chương 3 bản báo cáo |
| Nguyễn Hoàn Thiện | DisplayMenu  PrintDes  Tìm kiếm theo từ khóa  Chương 4 bản báo cáo |

**Lưu ý**: nhiệm vụ cần phân công rõ ràng, ko có kiểu là “cùng nhau” làm cái này !

# TÀI LIỆU THAM KHẢO

1. <https://www.howkteam.vn/>

2. <https://xuanthulab.net/>

3. [Tìm hiểu về cách thiết kế Class Diagram (viblo.asia)](https://viblo.asia/p/tim-hieu-ve-cach-thiet-ke-class-diagram-L4x5xLyY5BM?fbclid=IwAR170Q_LpgOihRFu-07ReE2wNFxaNukH8dsK8Swo6OQDCTZQ63bvStWMimU)