TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN TOÁN TỔ HỢP VÀ ĐỒ THỊ**

**Combinatorial Optimization**

*Người hướng dẫn*: **THẦY LÊ ĐÌNH THẬN**

*Người thực hiện*: **TRỊNH THẾ DUY– 51603083**

**LÊ TRÍ DŨNG – 51603065**

**PHƯƠNG LẬP PHÁT - 51403176**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 201**

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của TS Nguyễn Văn A;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trịnh Thế Duy*

*Lê Trí Dũng*

*Phương Lập Phát*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Ba thuật toán: Maximum Network Flow, Shortest Path và Minimum Spanning Tree

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc387692905)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN i](#_Toc387692906)

[TÓM TẮT i](#_Toc387692907)

[MỤC LỤC 1](#_Toc387692908)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 1](#_Toc387692909)

[CHƯƠNG 1 – MỞ ĐẦU 1](#_Toc387692910)

[1.1 Tiểu mục cấp 1 1](#_Toc387692911)

[1.1.1 Tiểu mục cấp 2 1](#_Toc387692912)

[1.1.1.1 Tiểu mục cấp 3 1](#_Toc387692913)

[1.1.1.2 Tiểu mục cấp 3 tiếp theo. 1](#_Toc387692914)

[1.1.2 Tiểu mục cấp 2 tiếp theo 1](#_Toc387692915)

[1.2 Nội dung của chương này 1](#_Toc387692916)

[CHƯƠNG 2 – TỔNG QUAN 1](#_Toc387692917)

[1.1 Trình bày công thức toán học 1](#_Toc387692918)

[1.2 Trình bày một hình vẽ, sơ đồ 1](#_Toc387692919)

[CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT / NGHIÊN CỨU THỰC NGHIỆM 1](#_Toc387692920)

[3.1 Chèn bảng: 1](#_Toc387692921)

[3.2 Viết tắt 1](#_Toc387692922)

[3.3 Trích dẫn 1](#_Toc387692923)

[3.3.1 Tài liệu tham khảo và cách trích dẫn 1](#_Toc387692924)

[3.3.2 Qui định của Khoa Công nghệ thông tin 1](#_Toc387692925)

CHƯƠNG 1 – Thuật toán

1.1 Maximum network flow

1.2 Shortest Path

* Shortest Path là bài toán tìm đường đi ngắn nhất từ đỉnh này tới đỉnh khác trong đồ thị.
* Thuật toán phổ biến nhất với bài toán này là thuật toàn Dijkstra.
* Thuật toán hoạt động như sau:
  + Input: một đồ thị có hướng được biểu diễn dưới dạng ma trận kề.
  + Khởi tạo mảng output chứa các đỉnh và khoảng cách từ đỉnh source tới đỉnh đó, một biến sptSet[] boolean để đánh dấu xem đỉnh đó đã được thuật toán duyệt qua chưa.
  + Ở bước lặp đầu tiên, thuật toán sẽ tìm đỉnh có khoảng cách ngắn nhất với đỉnh source, tiếp theo set sptSet[] = true.
  + Ờ vòng lặp thứ 2, cập nhật khoảng cách cho đỉnh đó tới source nếu như đỉnh đó có sptSet[] = false.

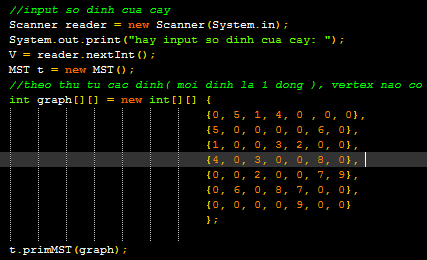
Code

|  |
| --- |
| import java.util.\*;  import java.lang.\*;  import java.io.\*;    class ShortestPath  {  //So dinh cua graph  static final int V=9;  int minDistance(int dist[], Boolean sptSet[])  {  //Khoi tao gia tri be nhat  int min = Integer.MAX\_VALUE, min\_index=-1;    for (int v = 0; v < V; v++)  if (sptSet[v] == false && dist[v] <= min)  {  min = dist[v];  min\_index = v;  }    return min\_index;  }    //Ham in kq  void printSolution(int dist[], int n)  {  System.out.println("Dinh Khoang cach toi source");  for (int i = 0; i < V; i++)  System.out.println(i+" "+dist[i]);  }    //Ham tim duong di ngan nhat theo thuat toan Dijkstra  void dijkstra(int graph[][], int src)  {  int dist[] = new int[V]; //Mang chua output    //sptSet[i] = true neu dinh i da duoc duyet va chua output  Boolean sptSet[] = new Boolean[V];    //Khoi tao moi khoang cach voi gia tri vo cuc va sptSet[i] = false  for (int i = 0; i < V; i++)  {  dist[i] = Integer.MAX\_VALUE;  sptSet[i] = false;  }    //Khoang cach tu source toi chinh no luon = 0  dist[src] = 0;    //Tim khoang cach ngan nhat  for (int count = 0; count < V-1; count++)  {  //Chon dinh co khoang cach ngan nhat toi source trong danh sach  //cac dinh chua duyet. u luon la source trong lan lap dau tien.  int u = minDistance(dist, sptSet);    //Danh dau dinh dang duyet  sptSet[u] = true;    //Update khoang cach moi cho dinh dang duyet  for (int v = 0; v < V; v++)    //Update dist[v] chi khi sptSet[] = false, co mot cach  //tu u den v, va khoang cach tu source toi v qua u  //nho hon khoang cach dist[v] hien tai  if (!sptSet[v] && graph[u][v]!=0 &&  dist[u] != Integer.MAX\_VALUE &&  dist[u]+graph[u][v] < dist[v])  dist[v] = dist[u] + graph[u][v];  }  printSolution(dist, V);  }    public static void main (String[] args)  {  int graph[][] = new int[][]{{0, 4, 0, 0, 0, 0, 0, 8, 0},  {4, 0, 8, 0, 0, 0, 0, 11, 0},  {0, 8, 0, 7, 0, 4, 0, 0, 2},  {0, 0, 7, 0, 9, 14, 0, 0, 0},  {0, 0, 0, 9, 0, 10, 0, 0, 0},  {0, 0, 4, 14, 10, 0, 2, 0, 0},  {0, 0, 0, 0, 0, 2, 0, 1, 6},  {8, 11, 0, 0, 0, 0, 1, 0, 7},  {0, 0, 2, 0, 0, 0, 6, 7, 0}  };  ShortestPath t = new ShortestPath();  Scanner reader = new Scanner(System.in);  System.out.print("Input source: ");  int s = reader.nextInt();  t.dijkstra(graph, s);  }  } |

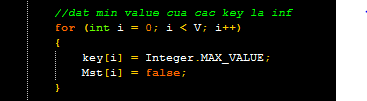
1.3 Minimum Spanning Tree

- Với thuật toán này ta nhận vào một đồ thị liên thông, vô hướng và cho ra 1 cây bao trùm nhỏ nhất với trọng số thấp nhất có thể. Ở đây nhóm em chọn sử dụng thuật toán Prim để thực hiện.

- Đầu tiên là tạo graph và nhập số đỉnh của graph ( biến V ).



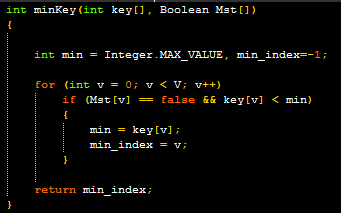
-Ban đầu tất cả các đỉnh sẽ có giá trị là infinite và đều chưa được xét.



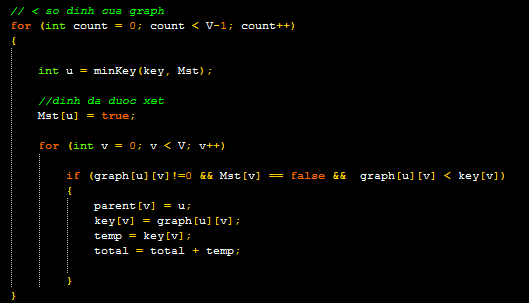
* Sau đó đặt đỉnh đầu tiên với giá trị = 0 để đỉnh được xét qua thuật toán ( là root của cây ).



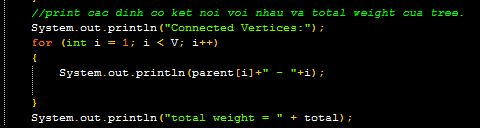
* Bắt đầu chạy thuật toán, đầu tiên sẽ là so sánh min valuekiểm tra xem đỉnh đã được xét hay chưa bằng function này



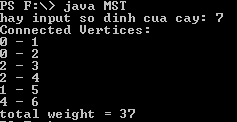
* Và kiểm tra xem đỉnh có cạnh kề hay không, đỉnh đã được xét hay chưa và xét giá trị của key để chọn key có giá trị nhỏ hơn.



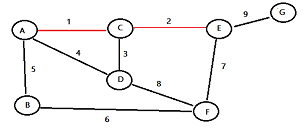
* Cuối cùng ta in biến total ( tổng weight của cây ) và các đỉnh có kết nối với nhau sau khi đã chạy MST.



* Đây là kết quả hiện ra



* Đây là graph input



* File code đã được kèm theo trong báo cáo với các dòng giải thích.