Họ tên Huỳnh Duy Khoa MSSV: 47.01.103.055

**Hướng dẫn cách dùng**

Ma trận kề có trọng số được nhập từ file DoThi.txt trong thư mục Debug của Project.

File gồm một dòng là số đỉnh (n) của đồ thị và ma trận nxn.

Eg:

2

0 1

0 0

Đỉnh bắt đầu là 0 và kết thúc là n-1.

**Bài 1: Breath Frist Search (BFS)**

**Input:** Ma trận kề (file.txt) và đỉnh bắt đầu (Console).

**Output:** Thứ tự các đỉnh sau khi duyệt.

Tư tưởng: là tìm kiếm ưu tiên chiều rộng hơn là chiều sâu. Thuật toán sẽ kiếm xung quanh để mở rộng trước khi đi xuống sâu hơn.

**Mã giả:**

BFS(u){

//Step 1: Khởi tạo

Queue = ; //Tạo một list rỗng

Push(queue,u);// Đẩy đỉnh u vào hàng đợi

Vistied[u] = true;//Dánh dấu là đỉnh u đã được thăm.

//Step 2: lặp khi mà hàng đợi vẫn còn phần tử

While (queue!= ∅) {

v = queue.front();// lấy đỉnh ở đầu hàng đợi

queue.pop();// xóa đỉnh đầu hàng đợi

<Thăm đỉnh v>;

//Duyệt các đỉnh kề với v mà chưa được thăm và đẩu vào hàng đợi

For(int x : kề [v]){

If(visited[x]){

Push(queue,x);

Visited[x] = true;

}

}

}}

**Bài 2: Depth First Search (DFS)**

Input: Ma trận kề (file.txt) và đỉnh bắt đầu (Console).

Output: Thứ tự các đỉnh sau khi duyệt.

Mã giả.

//Bắt đầu từ đỉnh u

DFS(u){

<Thăm đỉnh u>;

Visited[u]=true;

//Duyệt các đỉnh kề với đỉnh u

For (v : adj[u]){

If(!visited[v]){

DFS(v);

}

}

}

**Bài 3: Thuật toán Dijkstra**

Input: Ma trận trọng số (file.txt) và đỉnh bắt đầu và đỉnh kết thúc(Console).

Output: Đường đi ngắn nhất từ đỉnh start đến đỉnh end, và khoảng cách.

Mã giải:

Dijkstra (G(V,E),w,s):

For v 🡨 1 to V

D[u]🡨∞

d[s] 🡨 0

for every neighbor v of s

d[v] 🡨 w(s->v)

B🡨 V\{s}

Repeat

U🡨 argminx∊Bd[x]

B🡨 B\{u} <<d[u] = δ(s,u)>>

For every neighbor v of u

D[v] 🡨 min(d[v],d[u]+w(u 🡪 v))

Unitil B=φ

**Bài 4. Thuật toán Kruscal**

Input: Ma trận trọng số (file.txt)

Output: Thứ tự các cạch và trọng số được chọn.

Mã giả:

Bước 1: Khởi tạo:

MST=φ; E: tập cạnh

d(MST) = 0;

<bước 2> Sắp xếp

<sort các cạch ∊độ dài>

<bước 3> Lặp:

While(|MST|<n-1 and E !=φ}

e=E\{e}

if(MST U (e) không tạo thành chu trình){

MST=MST U e

d(MST) += d(e)

}

}

**Bài 5. Thuật toán Prim**

Input: Ma trận trọng số (file.txt) và đỉnh được chọn(console).

Output: Thứ tự các cạch và trọng số được chọn.

Mã giả:

Prim(u){

Bước 1: khởi tạo

Vmst={u};

V=V\{u};

MST = φ; d=0

Bước 2: Lặp

While (v≠φ){

e=(x,y) //cạnh ngắn nhất x ∊V, y ∊Vmst

MST =MST U(e)

d=d+d(e);

VMST=VMSTU{x};

V=V\{x};

B3: Trả về kết quả