**Buổi 09 CÂY KHUNG – Tìm cây khung theo thuật toán DFS**

**Tìm cây khung nhỏ nhất theo thuật toán KRUSKAL, PRIM**

**Bài 1** : Tìm cây khung theo thuật toán DFS

Cho đồ thị vô hướng liên thông 𝐺 = (𝑉, 𝐸) có 𝑛 đỉnh, các đỉnh được đánh số từ 1 đến 𝑛. Hãy tìm cây khung của đồ thị 𝐺 theo thuật toán DFS tại đỉnh 1

|  |  |  |
| --- | --- | --- |
| Đồ thị | Input: Bai1.txt | Output : tập các cạnh của cây khung |
|  | 11 13  0 1  0 2  1 2  1 3  3 4  3 5  3 8  4 6  4 9  5 7  6 9  7 8  9 10 | Cây khung với đỉnh xuất phát 0 :  (0,1)  (1,2)  (1,3)  (3,4)  (4,6)  (6,9)  (9,10)  (3,5)  (5,7)  (7,8) |

*Giải thuật* :

void SpanningTree\_DFS(int u)  
{  
 1. Đánh dấu u đã viếng thăm  
 2. Xét mọi đỉnh v chưa được viếng thăm và kề u, với từng đỉnh v đó  
 {  
 T = T ∪ {(u, v)}  
 DFS(v);  
 }  
}

*Cài đặt* :

* Đọc Bai1.txt 🡪 chuyển sang đồ thị danh sách kề (AdjList EdgeListToAdjList(string filePath) : đã làm trong buổi 01)
* Trong class AdjList.cs, khai báo thêm thành phần :

LinkedList<Tuple<int, int>> tree để mô tả cho tập T và viết hai phương thức :

// Tìm cây khung theo DFS : SpanningTree(int s) với đỉnh xuất phát là s

// Có sử dụng đệ qui : TreeDFS(int s)

public void SpanningTree(int s)

{ // Khởi tạo tree

// Khởi tạo và gán giá trị ban đầu cho visited

// Gọi đệ qui : TreeDFS(s)

// Xuất cây khung tree

}

public void TreeDFS(int s)

{ // Đánh dấu duyệt s

// for(u thuộc Kề(s))

// Nếu chưa duyệt u

{ // Định nghĩa cạnh e = (s,u)

// Đưa e vào tree

// gọi đệ qui TreeDFS(u)

}

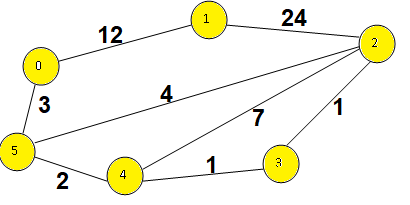
}

**Bài 2** : Tìm cây khung nhỏ nhất theo thuật toán KRUSKAL

|  |  |  |
| --- | --- | --- |
| Đồ thị | Input : Bai2.txt | Output : |
|  | 7 13  0 1 3  0 2 18  0 3 6  0 4 9  0 5 15  1 2 12  1 3 2  2 3 7  2 6 4  3 4 3  3 5 8  4 5 4  5 6 2 | Cây khung nhỏ nhất theo thuật toán Kruskal  (1,3) = 2  (5,6) = 2  (3,4) = 3  (0,1) = 3  (4,5) = 4  (2,6) = 4  Độ dài = 18 |

*Giải thuật* : (để đơn giản hơn, sử dụng đồ thị sau)

Gọi E : tập cạnh của G



T : lưu trữ tập cạnh cây khung của G

void Kruskal()

{

T = Ǿ ;

Gán nhãn khác nhau cho các đỉnh v thuộc V

Sắp thứ tự tập cạnh E tăng dần theo trọng số

For( cạnh(u,v) thuộc E)

Nếu nhãn(u) # nhãn(v)

{

T = T ∪ {cạnh(u,v)}

Đồng nhất nhãn cho u và v

}

}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| E | Các đỉnh | | | | | | T |
|  | 0 | 1 | 2 | 3 | 4 | 5 |  |
| (2,3) = 1 | 0 | 1 | 23 | | 4 | 5 | (2,3) = 1 |
| (3,4) = 1 | 0 | 1 | 234 | | | 5 | (3,4) = 1 |
| (4,5) = 2 | 0 | 1 | 2345 | | | | (4,5) =2 |
| (0,5) = 3 | 02345 | 1 | 02345 | | | | (0,5) = 3 |
| (2,5) = 4 |  |  |  |  |  |  |  |
| (2,4) = 7 |  |  |  |  |  |  |  |
| (0,1) = 12 | 012345 | | | | | | (0,1) = 12 |
| (1,2) = 24 |  |  |  |  |  |  |  |

*Cài đặt* : Sử dụng đồ thị danh sách cạnh WeightEdgeList.cs , trong đó :

* Khai báo các thành phần :

List<Tuple<int, int, int>> g;

int n; // số đỉnh của đồ thị

int m; // số cạnh của đồ thị

List<Tuple<int, int, int>> tree; // Cây khung nhỏ nhất spanningTree

bool[,] connected; // dùng cho giải thuật Kruskal

bool[] label; // dùng cho giải thuật Prim

// connected, label dùng đánh dấu các đỉnh liên thông

* Phương thức Kruskal tìm cây khung nhỏ nhất :

// Kruskal : tìm cây khung nhỏ nhất, kết quả lưu trong Tree

public void Kruskal()

{

// Sắp xếp các cạnh của g tăng dần theo trọng số (Item3)

g.Sort((x, y) => x.Item3.CompareTo(y.Item3));

// Khởi tạo connected[] (dùng dánh dấu các đỉnh cùng TPLT)

// Ban đầu, mỗi đỉnh là một thành phần liên thông **connected[i,j] = false** với mọi i, j

// Khởi tạo tree (tree = rỗng)

// Duyệt từng cạnh của g

{

// Nếu tree.Count = n - 1 🡪 Dừng (khi cây tree đủ n - 1 cạnh)

// Nếu 2 đỉnh cạnh e : e.Item1 và e.Item2 không liên thông

{ // Bỏ e vào tree

// Đồng nhất thành phần liên thông cho 2 đỉnh của cạnh e

}

}

// Xuất cây khung nhỏ nhất tree

Console.WriteLine(" Cây khung nhỏ nhất");

ShowTree();

}

* Phương thức xuất cây khung :

// Xuất cây khung cùng tổng trọng số các cạnh

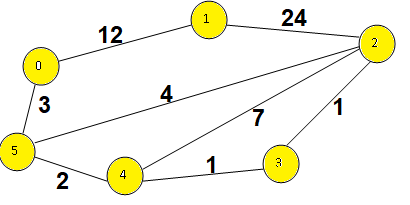
public void ShowTree()

{

}

**Bài 3** : Tìm cây khung nhỏ nhất theo thuật toán PRIM (đồ thị và input, output như bài 2)

*Thuật toán* :

Xây dựng tập đỉnh Vt và tập cạnh F của cây khung T

Void Prim(u)

{ (u : đỉnh bắt đầu)

F = ∅ ; Vt = {u} ;

while(số cạnh của F < n-1)

{

Chọn cạnh e = min{(u, v) / u ∈ Vt và v ∉ Vt};

F = F ∪ {e} ;

Vt = Vt ∪ {v} ;

}

}

Kết quả F lưu trữ cây khung nhỏ nhất

Để hiện thực thuật toán ta sẽ gán nhãn các đỉnh. Mỗi v ∉ Vt sẽ có 2 nhãn :

* d[v] = min {(v,u) / u ∈ Vt}
* near[v] = { u với (v,u) min }

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Vt | 1 | 2 | 3 | 4 | 5 | T |
| 0 (đỉnh XP) | 12, 0 | ∞ , 0 | ∞ , 0 | ∞ , 0 | 3, 0 (chọn) | (0,5) = 3 |
| 0 5 | 12, 0 | 4 , 5 | ∞ , 0 | 2 , 5 (chọn) |  | (5,4) = 2 |
| 0 5 4 | 12 , 0 | 4 , 5 | 1 , 4 (chọn) |  |  | (4,3) = 1 |
| 0 5 4 3 | 12 , 0 | 1 , 3 (chọn) |  |  |  | (3,2) = 1 |
| 0 5 4 3 2 | 12 , 0 (chọn) |  |  |  |  | (0,1) = 12 |
| 0 5 4 3 2 1 |  |  |  |  |  |  |

*Cài đặt* : Trong WeightEdgeList.cs viết các phương thức :

// Prim tìm cây khung nhỏ nhất, kết quả lưu trong tree. u là đỉnh xuất phát

public void Prim(int u)

{

// label đánh dấu các đỉnh cùng TPLT.

// Khởi tạo Label và gán label = false với mọi i (mỗi đỉnh là một TPLT)

// Đánh dấu u (bỏ u vào Vt)

label[u] = true;

// Khởi tạo một cạnh e(0, 0, 0) phục vụ cho giải thuật

// Khởi tạo tree (rỗng)

// Lặp khi tree.Count < n-1

{

// Chọn cạnh e = min{(u, v) / u thộc Vt và v thuộc Vt};

e = dmin();

// Bỏ e vào tree

// Đồnh nhất 2 nhãn của 2 đỉnh cạnh e : e.Item1 = e.Item2 = true;

}

// Xuất cây khung nhỏ nhất tree

ShowTree();

}

// Tìm cạnh e = min{(u, v) / u thuộc Vt và v thuộc Vt};

public Tuple<int, int, int> dmin()

{

// Khởi tạo cạnh emin phục vụ cho giải thuật

Tuple<int, int, int> emin = new Tuple<int, int, int>(int.MaxValue, int.MaxValue, int.MaxValue);

// Duyệt từng cạnh e trong g

// Nếu 2 đỉnh của e khác TPLT và e.Item3 < emin.Item3

// thay emin = e;

// trả về emin

}

**Bài 4** : Cây khung x

Cho đồ thị 𝐺 = (𝑉, 𝐸) có 𝑛 đỉnh (1 ≤ 𝑛 ≤ 105 ), các đỉnh được đánh số từ 1 đến 𝑛, 𝑚 cạnh (1 ≤ 𝑚 ≤ 105 ) và độ dài 𝑥. Hãy tìm cây khung nhỏ nhất có độ dài cạnh nhỏ nhất trong cây khung lớn hơn hay bằng 𝑥.

**Bài 5** : Cây khung e

Cho đồ thị 𝐺 = (𝑉, 𝐸) có 𝑛 đỉnh (1 ≤ 𝑛 ≤ 105 ), các đỉnh được đánh số từ 1 đến 𝑛, 𝑚 cạnh (1 ≤ 𝑚 ≤ 105 ) và cạnh e(v1,v2) cho trước. Hãy tìm cây khung nhỏ nhất có chứa cạnh e.