



Học viện Công nghệ Bưu chính Viễn thông  
Khoa Công nghệ thông tin 1

## Toán rời rạc 2

Cây và Cây khung của đồ thị

*Bộ môn KHMT, Khoa CNTT1*

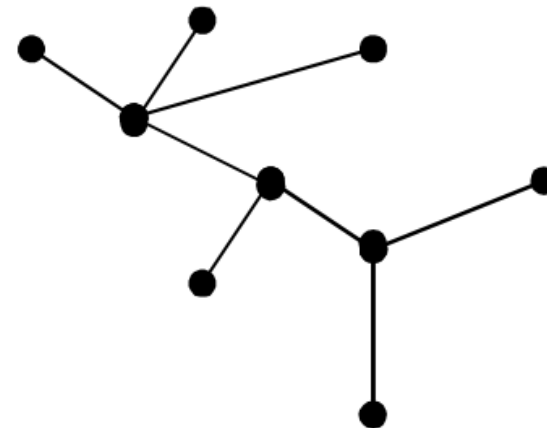
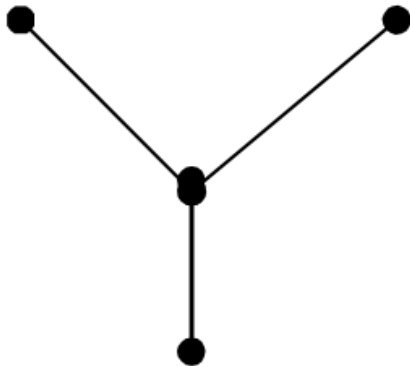


# Nội dung

- ▶ Cây và các tính chất của cây
- ▶ Cây khung của đồ thị
- ▶ Bài toán cây khung nhỏ nhất

# Định nghĩa và ví dụ

- ▶ **Định nghĩa 1:** Ta gọi **cây** là một đồ thị **vô hướng, liên thông, không có chu trình**
- ▶ **Định nghĩa 2:** Ta gọi **rừng** là một đồ thị **vô hướng, không có chu trình**
  - Như vậy rừng là một đồ thị mà mỗi thành phần liên thông của nó là một cây
- ▶ **Ví dụ:** một rừng có 3 cây



(Phuong ND, 2013)

## Các tính chất của cây

- ▶ **Định lý:** Giả sử  $T = \langle V, E \rangle$  là đồ thị vô hướng  $n$  đỉnh, khi đó những khẳng định sau là tương đương:
  - 1)  $T$  là một cây
  - 2)  $T$  không có chu trình và có  $n - 1$  cạnh
  - 3)  $T$  liên thông và có đúng  $n - 1$  cạnh
  - 4)  $T$  liên thông và mỗi cạnh của nó đều là cầu
  - 5) Giữa hai đỉnh bất kỳ của  $T$  được nối với nhau bởi đúng một đường đi đơn
  - 6)  $T$  không chứa chu trình nhưng hễ cứ thêm vào nó một cạnh ta thu được đúng một chu trình
  
- ▶ **Chứng minh:** Theo sơ đồ
$$(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) \Rightarrow (5) \Rightarrow (6) \Rightarrow (1)$$

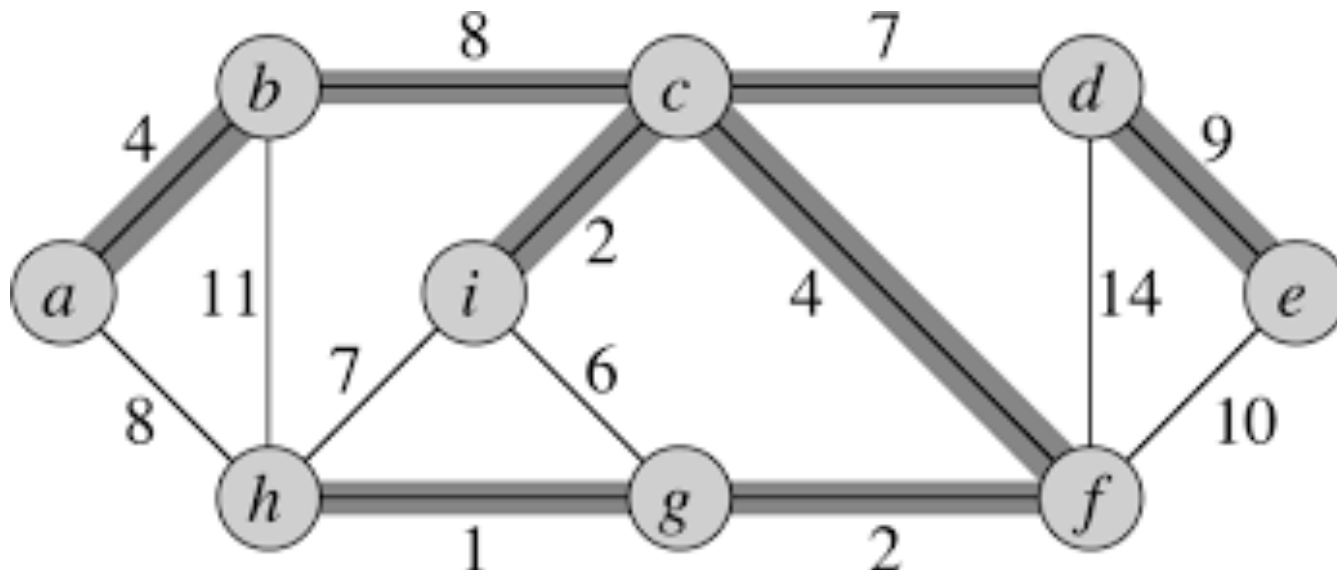


- ▶ 5

## Định nghĩa và ví dụ

- **Định nghĩa 3:** Cho  $G$  là đồ thị vô hướng liên thông. Ta gọi đồ thị con  $T$  của  $G$  là một **cây khung** của  $G$  (**Cây bao trùm**) nếu  $T$  thoả mãn hai điều kiện:
- $T$  là một cây
  - Tập đỉnh của  $T$  bằng tập đỉnh của  $G$

► **Ví dụ:**



# Xây dựng cây khung của đồ thị

---

- ▶ **Bài toán:** Cho đồ thị vô hướng  $G = \langle V, E \rangle$ . Hãy xây dựng một cây khung của đồ thị bắt đầu tại đỉnh  $u \in V$ .
  
- ▶ **Cách làm**
  - Sử dụng thuật toán duyệt DFS hoặc BFS
  - Mỗi khi ta đến được đỉnh  $v$  (tức  $chuaxet[v] = true$ ) từ đỉnh  $u$  thì cạnh  $(u, v)$  được kết nạp vào cây khung

# Xây dựng cây khung của đồ thị sử dụng thuật toán DFS (1/2)

## Thuật toán tạo cây khung từ một đỉnh $u$

```
Tree-DFS( $u$ ){  
     $chuaxet[u] = false$ ; //đánh dấu đỉnh  $u$  đã duyệt  
    for( $v \in Ke(u)$ ){  
        if(  $chuaxet[v]$ ){ //nếu  $v$  chưa được duyệt  
             $T = T \cup \{(u, v)\}$ ; //hợp cạnh  $(u, v)$  vào cây  
            Tree-DFS( $v$ ); //duyet theo chiều sâu từ  $v$   
        }  
    }  
}
```

khung



# Xây dựng cây khung của đồ thị sử dụng thuật toán DFS (2/2)

## Thuật toán xây dựng cây khung

```
Tree-Graph-DFS( ) {  
    // Khởi tạo các đỉnh đều chưa xét  
    for( $u \in V$ )  
         $chuaxet[u] = true$ ;  
  
     $root = \langle \text{đỉnh bất kỳ của đồ thị} \rangle$ ; // Lấy một đỉnh bất kỳ làm gốc  
     $T = \emptyset$ ; // Cây ban đầu chưa có cạnh nào  
    Tree-DFS( $root$ ); // Gọi thuật toán tạo cây khung từ một đỉnh  
  
    if( $|T| < n - 1$ )  
         $\langle \text{đồ thị không liên thông} \rangle$ ;  
    else  
         $\langle \text{ghi nhận tập cạnh của cây khung } T \rangle$ ;  
}
```

## Kiểm nghiệm thuật toán (1/2)

- Cho đồ thị vô hướng được biểu diễn bằng ma trận kề như hình bên. Áp dụng thuật toán xây dựng cây khung của đồ thị sử dụng DFS cho đồ thị trên bắt đầu từ đỉnh  $u = 1$ .

0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1	1	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	0
0	0	0	0	1	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	1	1	1	0

(Phuong ND, 2013)

# Kiểm nghiệm thuật toán (2/2)

Bước c	Ngăn xếp đỉnh theo thứ tự gọi Tree-DFS(u)	T
0	1	$T = \emptyset$
1	1, 2	$T = T \cup \{(1,2)\}$
2	1, 2, 3	$T = T \cup \{(2,3)\}$
3	1, 2, 3, 4	$T = T \cup \{(3,4)\}$
4	1, 2, 3	
5	1, 2, 3, 5	$T = T \cup \{(3,5)\}$
6	1, 2, 3, 5, 6	$T = T \cup \{(5,6)\}$
7	1, 2, 3, 5, 6, 7	$T = T \cup \{(6,7)\}$
8	1, 2, 3, 5, 6, 7, 8	$T = T \cup \{(7,8)\}$
9	1, 2, 3, 5, 6, 7, 8, 9	$T = T \cup \{(8,9)\}$
10	1, 2, 3, 5, 6, 7, 8, 9, 10	$T = T \cup \{(9,10)\}$
11	1, 2, 3, 5, 6, 7, 8, 9, 10, 11	$T = T \cup \{(10,11)\}$
12	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12	$T = T \cup \{(11,12)\}$
13	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13	$T = T \cup \{(12,13)\}$
Không thêm được cạnh nào nữa vào T		
$T = \{(1,2), (2,3), (3,4), (3,5), (5,6), (6,7), (7,8), (8,9),$ $(9,10), (10,11), (11,12), (12,13)\}$		

# Xây dựng cây khung của đồ thị sử dụng thuật toán BFS

**Tree-BFS**( $u$ ) {

*Bước 1: Khởi tạo*

$T = \emptyset$ ;  $queue = \emptyset$ ;  $push(queue, u)$ ;  $chuaxet[u] = false$ ;

*Bước 2: Lặp*

**while**( $queue \neq \emptyset$ ) {

$s = pop(queue)$ ;

**for**( $t \in Ke(s)$ ) {

**if**(  $chuaxet[t]$ ) {

$push(queue, t)$ ;

$T = T \cup \{(s, t)\}$ ;

$chuaxet[t] = false$ ;

        }

    }

}

*Bước 3: Trả lại kết quả*

**if**( $|T| < n - 1$ ) <đồ thị không liên thông>;

**else** <ghi nhận tập cạnh của cây khung  $T$ >;

}

# Kiểm nghiệm thuật toán (1/2)

- Cho đồ thị **vô hướng** được biểu diễn bằng ma trận kề như hình bên. Áp dụng thuật toán **xây dựng cây khung của đồ thị sử dụng BFS** cho đồ thị trên bắt đầu từ đỉnh  **$u = 1$** .

0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1	1	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	0
0	0	0	0	1	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	1	1	1
0	0	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	1	1	1	0

(Phuong ND, 2013)

# Kiểm nghiệm thuật toán (2/2)

Bước c	Trạng thái hàng đợi	T
0	1	$T = \emptyset$
1	2, 3, 4	$T = T \cup \{(1,2), (1,3), (1,4)\}$
2	3, 4	
3	4, 5	$T = T \cup \{(3,5)\}$
4	5	
5	6, 7, 8, 9	$T = T \cup \{(5,6), (5,7), (5,8), (5,9)\}$
6	7, 8, 9	
7	8, 9	
8	9	
9	10	$T = T \cup \{(9,10)\}$
10	11, 12, 13	$T = T \cup \{(10,11), (10,12), (10,13)\}$
11	12, 13	
12	13	
13	$\emptyset$	
$T = \{(1,2), (1,3), (1,4), (3,5), (5,6), (5,7), (5,8), (5,9), (9,10), (10,11), (10,12), (10,13)\}$		

# Nội dung

- ▶ Cây và các tính chất của cây
- ▶ Cây khung của đồ thị
- ▶ Bài toán cây khung nhỏ nhất

## Phát biểu bài toán

- ▶ Cho  $G = \langle V, E \rangle$  là đồ thị **vô hướng liên thông** với tập đỉnh  $V$  và tập cạnh  $E$ . Mỗi cạnh  $e$  của đồ thị được gán với một số không âm  $c(e)$  được gọi là độ dài cạnh.
- ▶ Giả sử  $H = \langle V, T \rangle$  là một cây khung của đồ thị  $G$ . Ta gọi độ dài  $c(H)$  của cây khung  $H$  là tổng độ dài các cạnh:

$$c(H) = \sum_{e \in T} c(e)$$

- ▶ **Bài toán:** Trong số các cây khung của đồ thị hãy tìm **cây khung có độ dài nhỏ nhất**.



## Ví dụ

---

- ▶ Bài toán nối mạng máy tính
  - Một mạng máy tính gồm  $n$  máy tính được đánh số từ  $1, 2, \dots, n$ . Biết chi phí nối máy  $i$  với máy  $j$  là  $c[i, j]$ ,  $i, j = 1, 2, \dots, n$ . Hãy tìm cách nối mạng sao cho chi phí là nhỏ nhất.
  
- ▶ Bài toán xây dựng hệ thống cáp
  - Giả sử ta muốn xây dựng một hệ thống cáp điện thoại nối  $n$  điểm của một mạng viễn thông sao cho điểm bất kỳ nào trong mạng đều có đường truyền tin tới các điểm khác. Biết chi phí xây dựng hệ thống cáp từ điểm  $i$  đến điểm  $j$  là  $c[i, j]$ . Hãy tìm cách xây dựng hệ thống mạng cáp sao cho chi phí là nhỏ nhất.

# Thuật toán Kruskal (1 / 2)

---

- ▶ Thêm dần từng cạnh vào cây khung
- ▶ Mỗi bước chọn cạnh có **trọng số nhỏ nhất chưa nằm trong cây khung**
  - Nếu việc thêm cạnh này vào cây **khung không tạo thành chu trình** thì thêm cạnh này vào
- ▶ Thuật toán dừng lại khi
  - Cây khung có đủ  $(n - 1)$  cạnh,
  - Hoặc không còn cạnh nào chưa nằm trong cây khung

## Thuật toán Kruskal (2/2)

**Kruskal**( ) {

**Bước 1 (khởi tạo):**

$T = \emptyset$ ; //Ban đầu tập cạnh cây khung là rỗng

$d(H) = 0$ ; //Ban đầu độ dài cây khung là 0

**Bước 2 (sắp xếp):**

<sắp xếp các cạnh đồ thị theo thứ tự tăng dần của trọng số>;

**Bước 3 (lặp):**

**while**( $|T| < n - 1 \ \&\& \ E \neq \emptyset$ ) {

$e = \text{<Cạnh có độ dài nhỏ nhất>};$

$E = E \setminus \{e\}$ ; //Loại cạnh  $e$  ra khỏi tập cạnh

**if** ( $T \cup \{e\}$  không tạo nên chu trình) {

$T = T \cup \{e\}$ ; //Đưa  $e$  vào cây khung

$d(H) = d(H) + d(e)$ ; //cập nhật độ dài cây khung

}

}

**Bước 4 (trả lại kết quả):**

**if**( $|T| < n - 1$ ) <Đồ thị không liên thông>;

**else return** ( $T, d(H)$ );

}

# Kiểm nghiệm thuật toán

- ▶ Áp dụng thuật toán **Kruskal** tìm cây khung nhỏ nhất cho đồ thị được biểu diễn bằng ma trận trọng số như hình bên ?

∞	2	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	∞	2	∞	∞	5	5	∞	∞	∞	∞	∞	∞
1	2	∞	4	∞	5	∞	∞	∞	∞	∞	∞	∞
3	∞	4	∞	5	5	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	5	∞	6	∞	∞	∞	6	∞	∞	∞
∞	5	5	5	6	∞	6	6	6	6	∞	∞	∞
∞	5	∞	∞	∞	6	∞	6	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	6	6	∞	7	∞	∞	7	7
∞	∞	∞	∞	∞	6	∞	7	∞	7	7	∞	∞
∞	∞	∞	∞	6	6	∞	∞	7	∞	7	7	∞
∞	∞	∞	∞	∞	∞	∞	∞	7	7	∞	8	∞
∞	∞	∞	∞	∞	∞	∞	7	∞	7	8	∞	8
∞	∞	∞	∞	∞	∞	∞	7	∞	∞	∞	8	∞

(Phuong ND, 2013)

## Thuật toán Prim (1/2)

- ▶ Duy trì hai tập đỉnh  $V_H$  (tập đỉnh của cây khung)  $V$  (tập các đỉnh chưa nằm trong cây khung)
  - Ban đầu  $V_H = \{s\}$ ,  $s$  là một đỉnh bất kỳ của đồ thị
  - $V$  bằng tập đỉnh của đồ thị trừ đi  $s$
- ▶ Mỗi bước chọn cạnh có **trọng số nhỏ nhất và có 1 đỉnh trong  $V_H$  và 1 đỉnh trong  $V$** 
  - Đưa cạnh này vào cây khung
  - Đưa đỉnh liền kề với cạnh này từ  $V$  sang  $V_H$
- ▶ Thuật toán dừng lại khi
  - Cây khung có đủ  $(n - 1)$  cạnh,
  - Hoặc không còn đỉnh nào trong  $V$

## Thuật toán Prim (2/2)

**Prim**(  $s$  ){

**Bước 1 (khởi tạo):**

$V_H = \{s\};$  //Ban đầu  $V_H$  chỉ chứa  $s$

$V = V \setminus \{s\};$  //Loại  $s$  ra khỏi  $V$

$T = \emptyset;$  //Cây khung ban đầu chưa có cạnh nào

$d(H) = 0;$  //Độ dài cây khung ban đầu bằng 0

**Bước 2 (lặp):**

**while**( $V \neq \emptyset$  ){

$e = (u, v);$  //Cạnh có độ dài nhỏ nhất với  $u \in V, v \in V_H$

**if**(không tìm được  $e$ )

**return** <Đồ thị không liên thông>;

$T = T \cup \{e\};$  //Đưa  $e$  vào cây khung

$d(H) = d(H) + d(e);$  //Cập nhật độ dài cây khung

$V_H = V_H \cup \{u\};$  //Đưa  $u$  vào  $V_H$

$V = V \setminus \{u\};$  //Loại  $u$  ra khỏi  $V$

}

**Bước 3 (trả lại kết quả):**

**return** ( $T, d(H)$ );

}

# Kiểm nghiệm thuật toán

- ▶ Áp dụng thuật toán **Prim** tìm cây khung nhỏ nhất cho đồ thị được biểu diễn bằng ma trận trọng số như hình bên bắt đầu từ đỉnh số 1?

∞	2	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	∞	2	∞	∞	5	5	∞	∞	∞	∞	∞	∞
1	2	∞	4	∞	5	∞	∞	∞	∞	∞	∞	∞
3	∞	4	∞	5	5	∞	∞	∞	∞	∞	∞	∞
∞	∞	∞	5	∞	6	∞	∞	∞	6	∞	∞	∞
∞	5	5	5	6	∞	6	6	6	6	∞	∞	∞
∞	5	∞	∞	∞	6	∞	6	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	6	6	∞	7	∞	∞	7	7
∞	∞	∞	∞	∞	6	∞	7	∞	7	7	∞	∞
∞	∞	∞	∞	6	6	∞	∞	7	∞	7	7	∞
∞	∞	∞	∞	∞	∞	∞	∞	7	7	∞	8	∞
∞	∞	∞	∞	∞	∞	∞	7	∞	7	8	∞	8
∞	∞	∞	∞	∞	∞	∞	7	∞	∞	∞	8	∞

(Phuong ND, 2013)

# Tóm tắt

---

- ▶ Khái niệm cây, các tính chất của cây
- ▶ Cây khung của đồ thị
  - Mọi đồ thị vô hướng liên thông đều có ít nhất một cây khung
  - Xây dựng cây khung của đồ thị sử dụng các thuật toán BFS và DFS
- ▶ Bài toán cây khung nhỏ nhất
  - Thuật toán Kruskal và thuật toán Prim





- Làm một số bài tập trong giáo trình