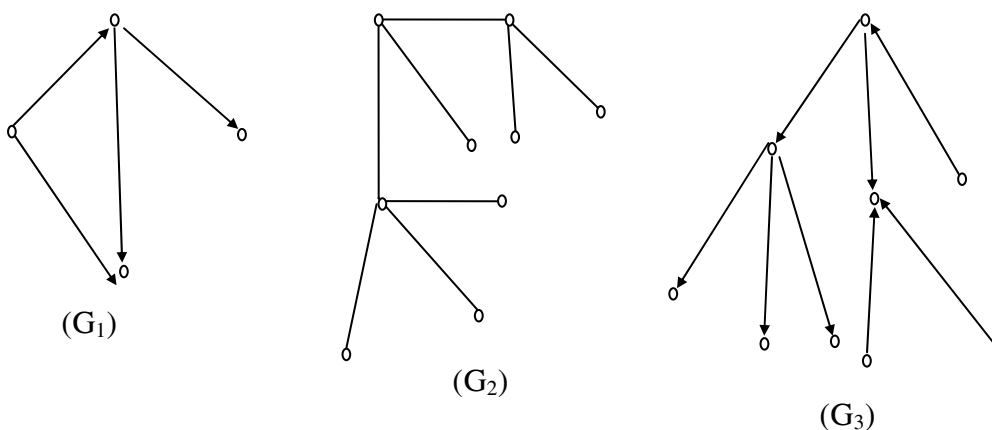


CHƯƠNG II. CÂY

II.1 Định nghĩa

- (a) Cây là đồ thị liên thông và không có chu trình
- (b) Một rừng p cây là một đồ thị gồm p thành phần liên thông, trong đó mỗi thành phần liên thông là một cây

Ví dụ. Trong các đồ thị dưới đây thì (G_1) không là cây, (G_2) và (G_3) là cây (chú ý định nghĩa chu trình của đồ thị có hướng trong chương I)



Ghi chú.

Định nghĩa cây hàm ý rằng mọi cây đều không chứa khuyên cũng không chứa cạnh song song.

II.2 Định lý (về sự tồn tại các đỉnh treo)

Nếu một cây T gồm n đỉnh với $n \geq 2$ thì T chứa ít nhất hai đỉnh treo

II.3 Định lý (về các định nghĩa tương đương)

Xét một đồ thị G gồm n đỉnh, các điều sau đây tương đương.

- (a) Đồ thị G là cây.
- (b) Giữa hai đỉnh bất kỳ của G , tồn tại duy nhất một dây chuyền nối chúng với nhau.
- (c) G liên thông tối thiểu (nghĩa là G liên thông và nếu xóa đi bất kỳ một cạnh nào của G thì nó không còn liên thông nữa).
- (d) Thêm một cạnh nối 2 đỉnh bất kỳ của G thì G sẽ chứa một chu trình duy nhất.
- (e) G liên thông và có $n-1$ cạnh
- (f) G không có chu trình và có $n-1$ cạnh

II.4 Cây tối đại (cây phủ, cây bao trùm, cây khung)

II.4.1 Định nghĩa

Cho $G=(X, E)$ là một đồ thị liên thông và $T=(X, F)$ là một đồ thị bộ phận của G . Nếu T là cây thì T được gọi là một cây tối đại của G .

II.4.2 Định lý (sự tồn tại cây tối đại)

Mọi đồ thị liên thông đều có chứa ít nhất một cây tối đại

II.4.3 Thuật toán (tìm một cây tối đại của đồ thị G)

Cho $G=(X, E)$ là một đồ thị liên thông gồm n đỉnh. Thuật toán sau đây cho phép tìm ra được một cây tối đại của G .

Bước 1. Chọn tùy ý $v \in X$ và khởi tạo $V := \{ v \}$; $T := \emptyset$
Bước 2. Chọn $w \in X \setminus V$ sao cho có một cạnh e nào đó của G nối w với một đỉnh trong V
Bước 3. Gán $V := V \cup \{ w \}$ và $T := T \cup \{ e \}$
Bước 4. Nếu T đủ $n-1$ phần tử thì dừng, ngược lại làm tiếp tục bước 2.

II.5 Cây tối đại ngắn nhất

II.5.1 Định nghĩa

Cho đồ thị $G=(X, E)$.

(a) Đồ thị G được gọi là có trọng nếu mỗi cạnh của G được tương ứng với một số thực dương, nghĩa là có một ánh xạ như sau:

$$\begin{aligned} L: E &\longrightarrow \mathbb{R}^+ \\ e &\longmapsto L(e) \end{aligned}$$

(b) Trọng lượng (hay giá) của một cây tối đại T trong đồ thị liên thông có trọng bằng với tổng trọng lượng các cạnh trong cây:

$$L(T) = \sum_{(e \in T)} L(e)$$

(c) Giả sử G liên thông có trọng. Cây tối đại ngắn nhất của G là cây tối đại có trọng lượng nhỏ nhất khi xét trong tập hợp tất cả các cây tối đại có thể có của G .

II.5.2 Thuật toán Prim

Cho $G=(X, E)$ là một đồ thị liên thông có trọng gồm n đỉnh. Thuật toán Prim được dùng để tìm ra cây tối đại ngắn nhất của G .

Bước 1. Chọn tùy ý $v \in X$ và khởi tạo $Y := \{ v \}$; $T := \emptyset$
Bước 2. Trong số những cạnh e nối đỉnh w với v mà $w \in X \setminus Y$ và $v \in Y$ ta chọn cạnh có trọng lượng nhỏ nhất.
Bước 3. Gán $Y := Y \cup \{ w \}$ và $T := T \cup \{ e \}$
Bước 4. Nếu T đủ $n-1$ phần tử thì dừng, ngược lại làm tiếp tục bước 2.

* Cài đặt thuật toán Prim.

Trong các thuật toán tìm cây tối đại ngắn nhất chúng ta có thể bỏ đi hướng các cạnh và các khuyên; đối với các cạnh song song thì có thể bỏ đi và chỉ để lại một cạnh trọng lượng nhỏ nhất trong chúng. Vì vậy dữ liệu nhập cho thuật toán thường là ma trận trọng lượng được qui ước như sau:

$$L_{ij} = \begin{cases} \text{trọng lượng cạnh nhỏ nhất nối } i \text{ đến } j \text{ nếu có} \\ 0 \text{ nếu không có cạnh nối } i \text{ đến } j \end{cases}$$

Ma trận có thể L được lưu trữ bằng một mảng 2 chiều trong bộ nhớ. Thủ tục Prim dưới đây được cài bằng ngôn ngữ Pascal, nhận vào tham biến G có kiểu DOTHI với giả thiết là số đỉnh G.n và ma trận G.L đã được đọc sẵn vào bộ nhớ

```
const MAXV=20;
type
  CANH=record
    dinh1, dinh2: byte;
    dodai: real;
  end;
  DOTHI=record
    n: byte;
    L: array[1..MAXV, 1..MAXV] of real;
    Y, X: set of 1..MAXV;
    T: array[1..MAXV] of CANH; {cay T}
    nT: byte; {so canh cua cay T}
    lT: real; {trong kuong cay T}
  end;
procedure Prim(var G: DOTHI);
var min: real;
    x, y, x0, y0: 1..MAXV;
begin
  G.nT := 0; G.lT := 0; G.X := [1..G.n]; G.Y := [1];
  while G.nT < G.n - 1 do
    begin
      min := -1;
      for x:=1 to g.n do
        for y:=1 to g.n do
          if (x in G.Y) and (y in (G.X - G.Y)) and
            (G.L[x,y]>0) then
            begin
              if (min = -1) or (min>G.L[x, y]) then
                begin
```

```

        min := G.L[x, y];
        x0 := x; y0 := y;
        end

    end;
    G.Y := G.Y + [y0];
    G.nT := G.nT + 1;    G.IT := G.IT + min;
    with G.T [G.nT] do
        begin
            dinh1 := x0;
            dinh2 := y0;
            dodai := min;
        end;
    end;
end;
end;
end;

```

II.5.3 Thuật toán Kruskal

Cho đồ thị $G=(X, E)$.

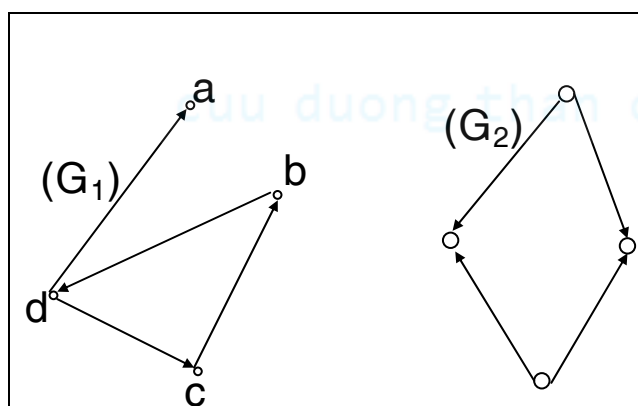
- Bước 1.** Sắp xếp các cạnh theo thứ tự trọng lượng tăng dần và khởi tạo $T := \emptyset$.
- Bước 2.** Lần lượt lấy từng cạnh e thuộc danh sách đã sắp xếp. Nếu $T+\{e\}$ không chứa chu trình thì gán $T := T+\{e\}$.
- Bước 3.** Nếu T đủ $n-1$ phần tử thì dừng, ngược lại làm tiếp tục bước 2.

II.6 Cây có hướng (cây ngoài)

II.6.1 Đồ thị có gốc

Cho $G=(X, E)$ là một đồ thị có hướng. Ta nói G là một đồ thị có gốc nếu tồn tại đỉnh $r \in X$ sao cho từ r có đường đi đến tất cả các đỉnh khác của đồ thị.
(Chú ý định nghĩa trong chương 1 về đường đi trong đồ thị có hướng.)

Ví dụ:



Trong đồ thị (G_1) , các đỉnh b, c, d đều là gốc; đỉnh a không phải là gốc. Đồ thị (G_2) không phải là đồ thị có gốc.

II.6.2 Đồ thị tựa liên thông mạnh

Cho $G=(X, E)$ là một đồ thị có hướng. Ta nói G là đồ thị tựa

liên thông mạnh nếu: với mọi đỉnh $i, j \in X$ luôn tồn tại một đỉnh $k \in X$ sao cho có đường đi từ k đến i và có đường đi từ k đến j .

Nhận xét:

Từ các định nghĩa ta suy ra ngay tính chất sau đối với một đồ thị có hướng:

Có gốc \Rightarrow Tựa liên thông mạnh \Rightarrow Liên thông.

Do tính chất hữu hạn của các đồ thị trong giáo trình này, chúng ta cũng có định lý sau đây.

Định lý.

G là đồ thị có gốc $\Leftrightarrow G$ là đồ thị tựa liên thông mạnh

II.6.3 Định nghĩa cây có hướng

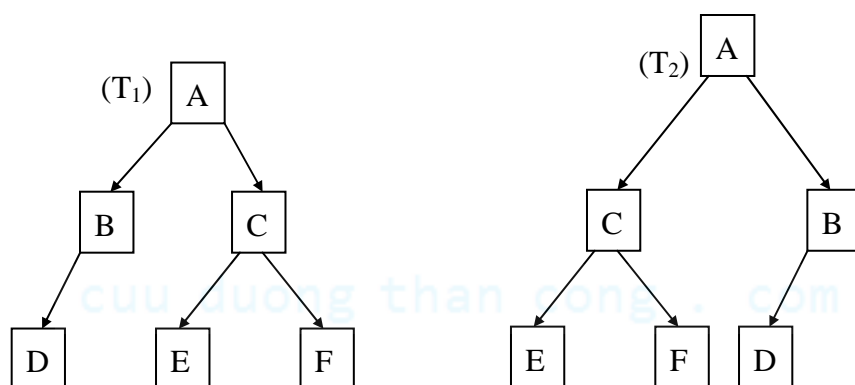
Cho $G=(X, E)$ là một đồ thị có hướng. G được gọi là cây có hướng nếu:

- (a) G không có chu trình,
- (b) G có gốc.

Ghi chú.

- Theo định nghĩa trong chương 1, một chu trình có thể không kể đến hướng của các cạnh.
- Từ định nghĩa ta suy ra cây có hướng cũng là cây.
- Khái niệm cây có hướng trong định nghĩa trên vẫn còn tổng quát hơn khái niệm cây trong các giáo trình tin học (chẳng hạn như giáo trình cấu trúc dữ liệu). Các cây trong các giáo trình tin học được vẽ ra giấy với nút gốc được vẽ trên cùng, nút cha luôn ở phía trên, ngoài ra phải có sự phân biệt giữa cây con bên trái và cây con bên phải.

Ví dụ



Hai cây có hướng (T_1) và (T_2) xem như đẳng cấu nhau trong giáo trình lý thuyết đồ thị nhưng chúng là hai cây hoàn toàn khác nhau trong giáo trình cấu trúc dữ liệu.

II.6.4 Định lý về các điều kiện tương đương với định nghĩa của cây có hướng

Cho $G=(X, E)$ là một đồ thị có hướng gồm n đỉnh. Các điều sau đây tương đương với nhau.

- (a) G là một cây có hướng.
- (b) G có một đỉnh r và từ r tồn tại một đường đi duy nhất đến tất cả các đỉnh còn lại.
- (c) G tựa liên thông mạnh tối tiểu (tức là nếu xóa bất kỳ một cạnh nào thì G sẽ không còn tựa liên thông mạnh).
- (d) G liên thông và có đỉnh r sao cho:
$$d^-(r)=0 \text{ và } d^-(i)=1, \forall i \in X \setminus \{r\}.$$
- (e) G không có chu trình và có đỉnh r sao cho:
$$d^-(r)=0 \text{ và } d^-(i)=1, \forall i \in X \setminus \{r\}.$$
- (f) G tựa liên thông mạnh và không có chu trình.
- (g) G tựa liên thông mạnh và có $n-1$ cạnh.

Ghi chú:

- Đỉnh r trong định lý trên là duy nhất và được gọi là gốc của cây có hướng.
- Mỗi đỉnh $i \in X, i \neq r$ do $d^-(i)=1$ nên có duy nhất một đỉnh j mà cạnh liên kết với (j, i) hướng vào i , đỉnh j được gọi đỉnh cha của i .
- Nếu đỉnh $x \in X$ thỏa điều kiện $d^+(x)=0$ thì x được gọi là lá của cây có hướng.

II.6.5 Sự tồn tại cây có hướng.

Cho G là đồ thị có hướng.

- (a) Nếu G có chứa một đồ thị bộ phận là cây có hướng thì G tựa liên thông mạnh.
- (b) Nếu G tựa liên thông mạnh thì G có chứa một đồ thị bộ phận là cây có hướng.

Ghi chú.

Nếu G tựa liên thông mạnh, T là một cây có hướng là đồ thị bộ phận G thì T cũng được gọi là cây có hướng tối đại của G .

II.6.6 Ma trận Kirchoff

(a) Định nghĩa

Cho $G=(X, E)$ là một đồ thị có hướng. Ta định nghĩa ma trận K như sau:

$$K_{ij} = \begin{cases} d^-(i) & \text{nếu } i=j \\ -B_{ij} & \text{nếu } i \neq j \end{cases}$$

(Trong đó B_{ij} là phần tử ở dòng i cột j của ma trận kề)

(b) Định lý (Kirchoff)

Giả sử G là đồ thị có hướng đơn, n đỉnh, $n-1$ cạnh có ma trận Kirchhoff là K . Gọi $K(1, 1)$ là ma trận có được từ ma trận K bằng cách bỏ đi dòng 1 và cột 1, khi đó G là cây ngoài có gốc tại đỉnh $1 \in X$ khi và chỉ khi $\det K(1, 1) = 1$.

cuu duong than cong . com

cuu duong than cong . com