Cây và cây khung của đồ thị

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

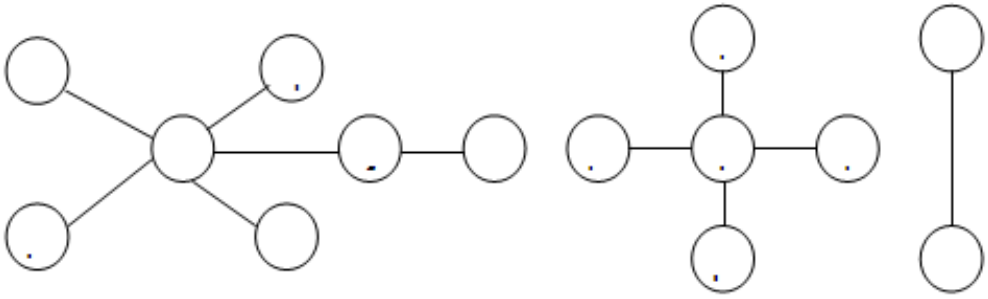
Cây và cây khung của đồ thị

Cây và các tính chất cơ bản của cây

Định nghĩa 1 Cây là một đồ thị vô hướng liên thông, không chứa chu trình và có ít nhất hai đỉnh.

Một đồ thị vô hướng không chứa chu trình và có ít nhất hai đỉnh gọi là một rừng. Trong một rừng, mỗi thành phần liên thông là một cây.

Ví dụ 1 Rừng sau có 3 cây:



Mệnh đề 1 Nếu T là một cây có n đỉnh thì T có ít nhất hai đỉnh treo.

Chứng minh: Lấy một cạnh (a,b) tuỳ ý của cây T. Trong tập hợp các đường đi sơ cấp chứa cạnh (a,b), ta lấy đường đi từ u đến v dài nhất. Vì T là một cây nên u ≠ v. Mặt khác, u và v phải là hai đỉnh treo, vì nếu một đỉnh, u chẳng hạn, không phải là đỉnh treo thì u phải là đầu mút của một cạnh (u,x), với x là đỉnh không thuộc đường đi từ u đến v. Do đó, đường đi sơ cấp từ x đến v, chứa cạnh (a,b), dài hơn đường đi từ u đến v, trái với tính chất đường đi từ u đến v đã chọn.

Định lý 2: Cho T là một đồ thị có n ≥ 2 đỉnh. Các điều sau là tương đương:

1) T là một cây.

2) T liên thông và có n?1 cạnh.

3) T không chứa chu trình và có n?1 cạnh.

4) T liên thông và mỗi cạnh là cầu.

5) Giữa hai đỉnh phân biệt bất kỳ của T luôn có duy nhất một đường đi sơ cấp.

6) T không chứa chu trình nhưng khi thêm một cạnh mới thì có được một chu trình duy nhất.

Chứng minh:1)⇒2) Chỉ cần chứng minh rằng một cây có n đỉnh thì có n?1 cạnh. Ta chứng minh bằng quy nạp. Điều này hiển nhiên khi n=2. Giả sử cây có k đỉnh thì có k?1 cạnh, ta chứng minh rằng cây T có k+1 đỉnh thì có k cạnh. Thật vậy, trong T nếu ta xoá một đỉnh treo và cạnh treo tương ứng thì đồ thị nhận được là một cây k đỉnh, cây này có k?1 cạnh, theo giả thiết quy nạp. Vậy cây T có k cạnh.

2)⇒3) Nếu T có chu trình thì bỏ đi một cạnh trong chu trình này thì T vẫn liên thông. Làm lại như thế cho đến khi trong T không còn chu trình nào mà vẫn liên thông, lúc đó ta được một cây có n đỉnh nhưng có ít hơn n?1 cạnh, trái với 2).

3)⇒4) Nếu T có k thành phần liên thông T_1, ..., T_k lần lượt có số đỉnh là n_1, ..., n_k (với $n_1+n_2+ \dots +n_k=n$) thì mỗi T_i là một cây nên nó có số cạnh là n_i?1. Vậy ta có

$$n-1=(n_1-1)+(n_2-1)+ \dots +(n_k-1)=(n_1+n_2+ \dots +n_k)-k=n-k.$$

Do đó k=1 hay T liên thông. Hơn nữa, khi bỏ đi một cạnh thì T hết liên thông, vì nếu còn liên thông thì T là một cây n đỉnh với n?2 cạnh, trái với điều đã chứng minh ở trên.

4)⇒5) Vì T liên thông nên giữa hai đỉnh phân biệt bất kỳ của T luôn có một đường đi sơ cấp, nhưng không thể được nối bởi hai đường đi sơ cấp vì nếu thế, hai đường đó sẽ tạo ra một chu trình và khi bỏ một cạnh thuộc chu trình này, T vẫn liên thông, trái với giả thiết.

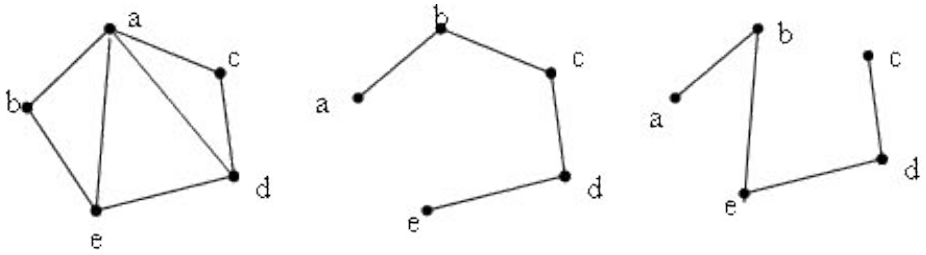
5)⇒6) Nếu T chứa một chu trình thì hai đỉnh bất kỳ trên chu trình này sẽ được nối bởi hai đường đi sơ cấp. Ngoài ra, khi thêm một cạnh mới (u,v), cạnh này sẽ tạo nên với đường đi sơ cấp duy nhất nối u và v một chu trình duy nhất.

6)⇒1) Nếu T không liên thông thì thêm một cạnh nối hai đỉnh ở hai thành phần liên thông khác nhau ta không nhận được một chu trình nào. Vậy T liên thông, do đó nó là một cây.

Cây khung của đồ thị

Định nghĩa 2. Giả sử G = (V, E) là đồ thị vô hướng liên thông. Cây T = (V, F) với F ⊂ E được gọi là **cây khung** của đồ thị G.

Ví dụ 2 Đồ thị G và cây khung của nó được cho trong hình 2



Hình 2 Đồ thị và các cây khung của nó.

Định lý sau đây cho biết số lượng cây khung của đồ thị đầy đủ K_n:

Định lý 3 (Cayley). *Số lượng cây khung của đồ thị K n là n n-2 .*

Định lý 3 cho thấy số lượng cây khung của đồ thị là một số rất lớn. Bây giờ ta xét áp dụng của thuật toán tìm kiếm theo chiều sâu và theo chiều rộng trên đồ thị để xây dựng cây khung của đồ thị vô hướng liên thông. Trong cả hai trường hợp mỗi khi ta đến được đỉnh mới u (tức Chuaxet[u]=true) từ đỉnh v thì cạnh (v, u) sẽ được kết nạp vào cây khung. Hai thuật toán tương ứng được trình bày trong hai thủ tục sau đây.

Procedure stree_DFS(v);

(tim kiem theo chieu sau ap dung vao tim tap canh cua cay khung T cua do thi vo huong lien thong G cho boi danh sach ke. Cac bien Chuaxet, Ke, T la toan cuc*)*

begin

Chuaxet[v]:=false;

For u ∈ Ke(v) do

If Chuaxet[u] then

Begin

T:=T ? (u,v);

STREE_DFS(u);

```
End;
end;
(* Main Program *)
begin
(* Initialition *)
for u ∈ V do Chuaxet[u]:=true;
T := Ø ; (* T la tap canh cua cay khung *)
STREE_DFS(root); ( root la dinh nao do cua do thi *)
end.
```

```
Procedure Stree_BFS(v);
(* tim kiem theo chieu rong ap dung tim tap canh cua cau khung T cua do thi vo huong
lien thong G cho boi danh sach Ke *)
begin
Queue:= Ø ;
Queue ⇐ r;
Chuaxet[r]:=false;
While queue <> Ø do
Begin
V ⇐ queue;
For r ∈ Ke(v) do
If Chuaxet[u] then
Begin
```

Queue ⇐ u;

Chuaxet[u]:=false;

T:= T ? (u,v);

End;

End;

end;

(* Main Program *);

begin

for u ∈ V do Chuaxet[u]:=true;

T := ∅ ; (T la tap canh cua cay khung *)*

Stree_BFS(root); (* root la mot dinh tuy y cua do thi *)

end.

Chú ý:

1. Lập luận tương tự như trong phần trước có thể chỉ ra được rằng các thuật toán mô tả ở trên có độ phức tạp tính toán O(m+n).

2. Cây khung tìm được theo thủ tục Stree_BFS() là cây đường đi ngắn nhất từ gốc r đến tất cả các đỉnh còn lại của đồ thị.

Xây dựng tập các chu trình cơ bản của đồ thị

Bài toán xây dựng cây khung của đồ thị liên quan chặt chẽ đến một số bài toán ứng dụng khác của lý thuyết đồ thị: bài toán xây dựng tập các chu trình cơ bản của đồ thị mà ta sẽ xét trong mục này.

Giả sử G=(V, E) là đơn đồ thị vô hướng liên thông, H=(V, T) là cây khung của nó. Các cạnh của đồ thị thuộc cây khung ta sẽ gọi là các cạnh trong, còn các cạnh còn lại sẽ gọi là cạnh ngoài.

Định nghĩa 3 Nếu thêm một cạnh ngoài e ∈ E\T vào cây khung H chúng ta sẽ thu được đúng một chu trình trong H, ký hiệu chu trình này là C_e. Tập các chu trình ? = { C_e : e ∈ E\T } được gọi là tập các chu trình cơ bản của đồ thị G.

Giả sử A và B là hai tập hợp, ta đưa vào phép toán sau

A ? B = (A ?B) \ (A ? B).

Tập A ? B được gọi là hiệu đối xứng của hai tập A và B.

Tên gọi chu trình cơ bản gắn liền với sự kiện là mỗi chu trình của đồ thị đều có thể thu được từ các chu trình cơ bản như chỉ ra trong định lý sau đây:

Định lý 3 Giả sử G=(V,E) là đồ thị vô hướng liên thông, H=(V,T) là cây khung của nó. Khi đó mọi chu trình của đồ thị G điều có thể biểu diễn như là hiệu đối xứng của một số các chu trình cơ bản.

Việc tìm tập hợp chu trình cơ bản giữ một vai trò quan trọng trong vấn đề giải tích mạng điện. Cụ thể hơn, theo mỗi chu trình cơ bản của đồ thị tương ứng với mạng điện cần phân tích ta sẽ thiết lập được một phương trình tuyến tính theo định luật Kirchoff: tổng hiệu điện thế dọc theo một mạch vòng là bằng không. Hệ thống phương trình tuyến tính thu được cho phép tính toán hiệu điện thế trên mọi đường dây của lưới điện.

Ta sẽ xây dựng thuật toán xây dựng các chu trình cơ bản dựa trên thủ tục tìm kiếm theo chiều sâu trên đồ thị. Thuật toán có cấu trúc tương tự như thuật toán xây dựng cây khung theo thủ tục tìm kiếm theo chiều sâu mô tả trong mục trước.

Thuật toán xây dựng tập các chu trình cơ bản.

Giả thiết rằng đồ thị G=(V,E) được mô tả bằng danh sách Ke(v),v∈ V.

Procedure Cycle(v);

(* tim kiem cac chu trinh co ban cua thanh phan lien thong chua dinh v; cac bien d, num , stack, index la bien toan cuc *)

begin

d:=d+1; stack[d]:=v; num:=num+1;index[v]:=num;

for u ∈ Ke(v) do

if index[u]=0 then cycle(u)

```
else
if (u <> stack[d-1]) and (index[v]>index[u]) then
<Ghi nhan chu trinh voi cac dinh:
stack[d], stack[d-1],. . ., stack[c], voi stack[c]=u>
d:=d-1;
end;
(* Main Program *)
begin
for v ∈ V do Index[v]:=0;
num:=0; d:=0; stack[0]:=0;
for v ∈ V do
if Index[v]=0 then cycle(v);
end.
```

Chú ý: Độ phức tạp tính toán của thuật toán vừa mô tả là O(|E| |V|).