

Duyệt cây & cây khung

Trần Vĩnh Đức

Ngày 7 tháng 2 năm 2017

Nội dung

1 Duyệt cây

- Hệ địa chỉ phổ dụng
- Các thuật toán duyệt cây
- Ký pháp trung tố, tiền tố, và hậu tố

2 Cây khung

- Mở đầu
- Tìm kiếm ưu tiên chiều sâu
- Tìm kiếm ưu tiên chiều rộng

3 Cây khung nhỏ nhất

Gán nhãn cho cây

- 1 Gốc có nhãn là 0

Gán nhãn cho cây

- 1 Gốc có nhãn là 0
- 2 các con của gốc được gán nhãn từ trái qua phải:

1, 2, ... k

Gán nhãn cho cây

- 1 Gốc có nhãn là 0
- 2 các con của gốc được gán nhãn từ trái qua phải:

$$1, 2, \dots k$$

- 3 Với mỗi đỉnh v ở mức n có nhãn A thì các con của nó được gán nhãn từ trái qua phải:

$$A.1, A.2, \dots, A.k$$

Gán nhãn cho cây

- 1 **Gốc** có nhãn là 0
- 2 các **con của gốc** được gán nhãn từ trái qua phải:

1, 2, ... k

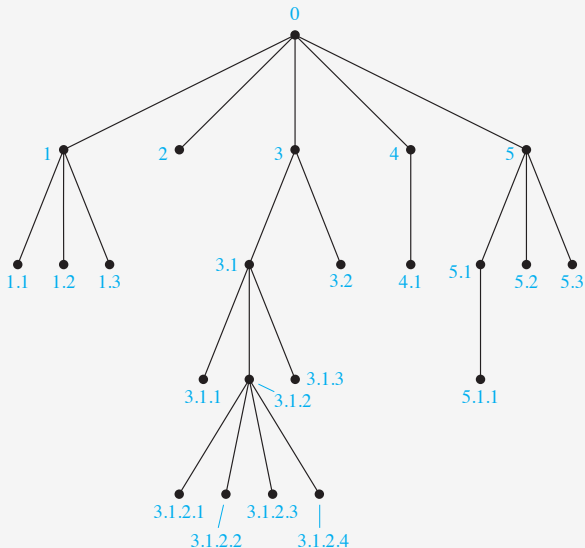
- 3 Với mỗi đỉnh v ở mức n có nhãn A thì các con của nó được gán nhãn từ trái qua phải:

A.1, A.2, ..., A.k

Thứ tự của các nhãn là thứ tự từ điển. Ví dụ

$$0 < 1 < 1.1 < 1.2 < 1.3 < 2 < 2.1 < 2.2 < 3 < 3.1$$

Hệ địa chỉ phổ dụng



Duyệt tiền thứ tự

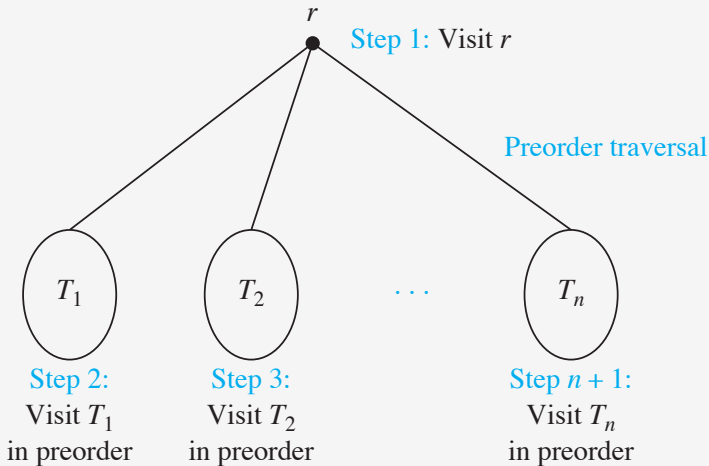
Định nghĩa

Giả sử T là một cây có gốc r .

- Nếu T chỉ có một đỉnh r thì r là cách **duyet tien thu tu** của T .
- Nếu T_1, T_2, \dots, T_n là các cây con tại r từ trái qua phải:

Duyệt tiền thứ tự sẽ thăm r đầu tiên.

Tiếp tục duyệt T_1 theo **tiền thứ tự**, sau đó duyệt T_2 theo **tiền thứ tự**, cứ vậy cho đến khi T_n được duyệt theo kiểu **tiền thứ tự**.



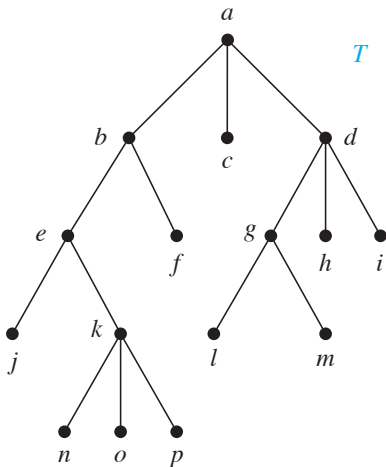
Hình: Duyệt tiền thứ tự

Thuật toán duyệt tiền thứ tự

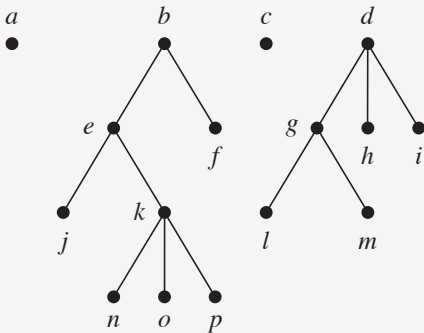
```
procedure preorder(Cây có gốc  $T$ )
|    $r =$  gốc của  $T$  ;
|   Thăm  $r$  ;
|   for mỗi con  $c$  của  $r$  tính từ trái qua phải do
|       |    $T(c) =$  cây con có gốc tại  $c$  ;
|       |   preorder ( $T(c)$ ) ;
|   end
```

Ví dụ

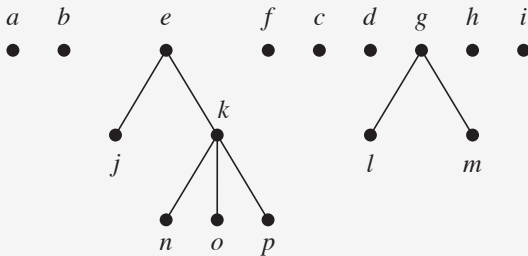
Hãy duyệt cây theo kiểu tiền thứ tự.



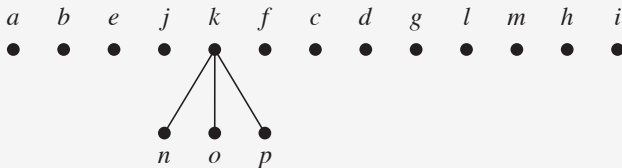
Ví dụ



Ví dụ



Ví dụ



Cuối cùng ta được:

$a, b, e, j, k, n, o, p, f, c, d, g, l, m, h, i$

Duyệt trung thứ tự

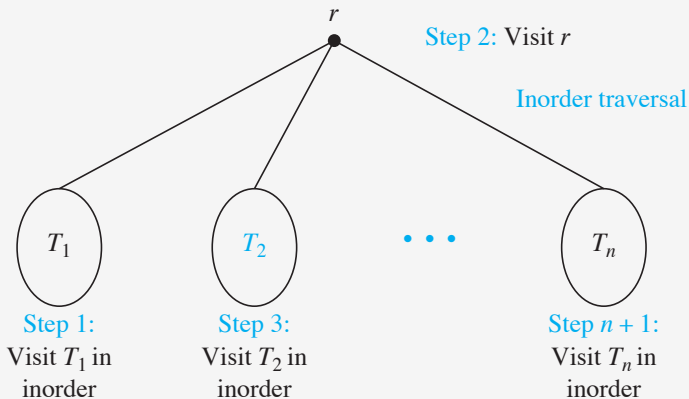
Định nghĩa

Giả sử T là một cây có gốc r .

- Nếu T chỉ có một đỉnh r thì r là cách **duyet trung thứ tự** của T .
- Nếu T_1, T_2, \dots, T_n là các cây con tại r từ trái qua phải:
Duyệt trung thứ tự sẽ bắt đầu bằng việc duyệt T_1 theo kiểu **trung thứ tự**,

sau đó viếng thăm r .

Tiếp tục duyệt T_2 theo **trung thứ tự**, tiếp tục duyệt T_3 theo kiểu **trung thứ tự**, cứ vậy cho đến khi T_n được duyệt theo kiểu **trung thứ tự**.



Hình: Duyệt trung thứ tự

Thuật toán duyệt trung thứ tự

procedure inorder(Cây có gốc T)

$r =$ gốc của T ;

if r là một lá **then**

 | Thăm r ;

end

else

$\ell =$ con đầu tiên của r tính từ trái qua phải;

$T(\ell) =$ cây con có gốc tại ℓ ;

 inorder ($T(\ell)$);

 Thăm r ;

for mỗi con $c \neq \ell$ của r tính từ trái qua phải **do**

 | $T(c) =$ cây con có gốc tại c ;

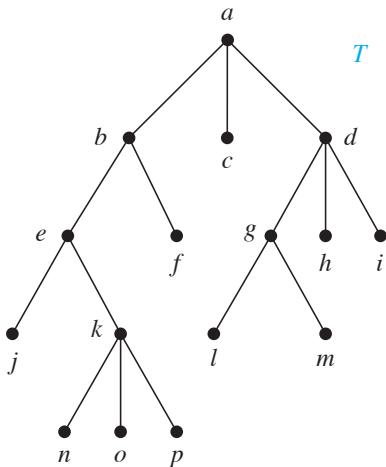
 | inorder ($T(c)$);

end

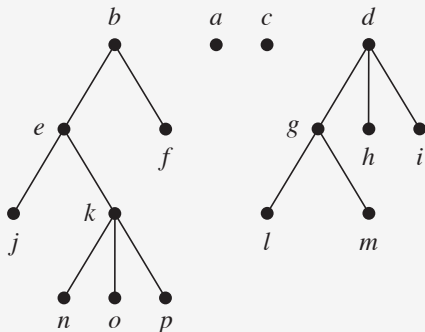
end

Ví dụ

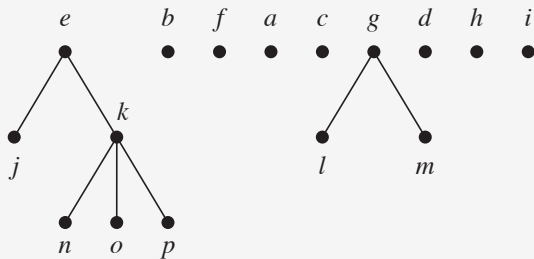
Hãy duyệt cây theo kiểu trung thứ tự.



Ví dụ: Duyệt trung thứ tự



Ví dụ: Duyệt trung thứ tự



Ví dụ: Duyệt trung thứ tự



Cuối cùng ta được:

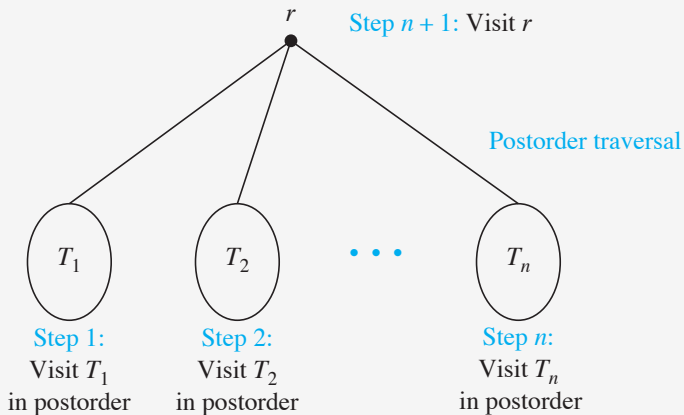
j, *e*, *n*, *k*, *o*, *p*, *b*, *f*, *a*, *c*, *l*, *g*, *m*, *d*, *h*, *i*

Duyệt hậu thứ tự

Định nghĩa

Giả sử T là một cây có gốc r .

- Nếu T chỉ có một đỉnh r thì r là cách duyệt hậu thứ tự của T .
- Nếu T_1, T_2, \dots, T_n là các cây con tại r từ trái qua phải:
Duyệt hậu thứ tự sẽ bắt đầu bằng việc duyệt T_1 theo kiểu hậu thứ tự, sau đó duyệt T_2 theo hậu thứ tự, và cứ tiếp tục cho đến khi T_n được duyệt theo kiểu hậu thứ tự,
và kết thúc bằng việc thăm r .



Hình: Duyệt hậu thứ tự

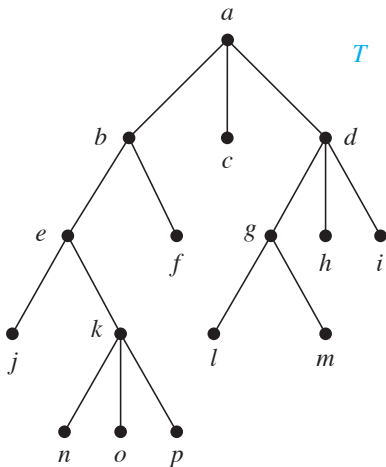
Thuật toán duyệt hậu thứ tự

```

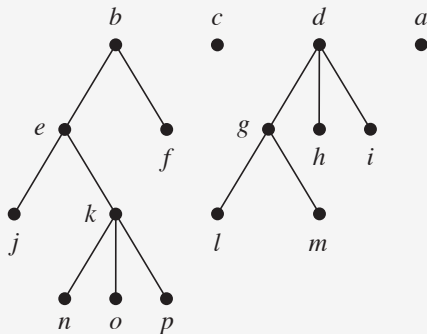
procedure postorder(Cây có gốc  $T$ )
     $r =$  gốc của  $T$ ;
    for mỗi con  $c$  của  $r$  tính từ trái qua phải do
        |    $T(c) =$  cây con có gốc tại  $c$ ;
        |   postorder ( $T(c)$ );
    end
    Thăm  $r$ ;
    
```


Ví dụ

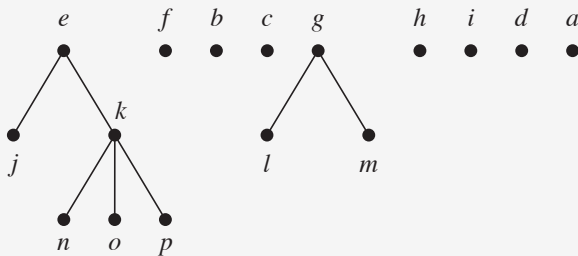
Hãy duyệt cây theo kiểu hậu thứ tự.



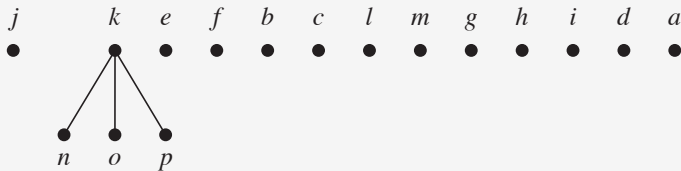
Ví dụ: Duyệt hậu thứ tự



Ví dụ: Duyệt hậu thứ tự



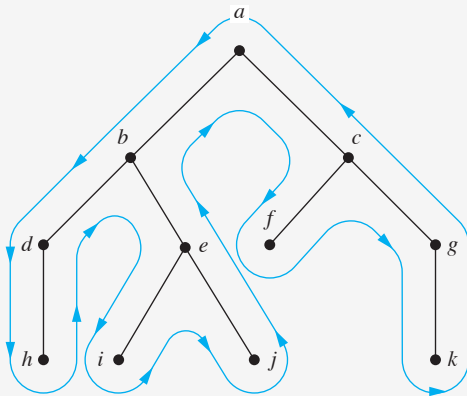
Ví dụ: Duyệt hậu thứ tự



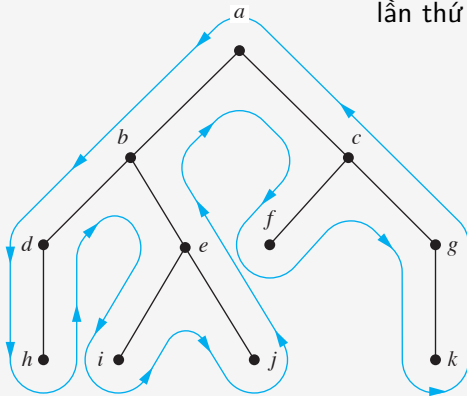
Cuối cùng ta được:

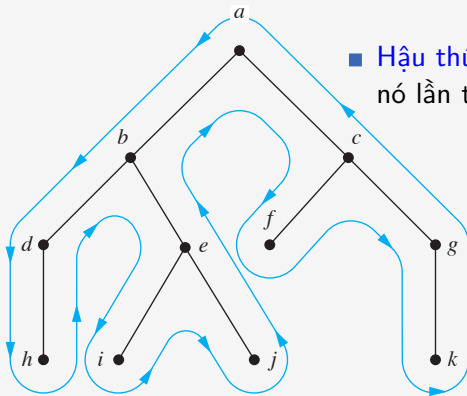
$j, n, o, p, k, e, f, b, c, l, m, g, h, i, d, a$

- **Tiền thứ tự:** Liệt kê mỗi đỉnh ngay khi đi qua nó



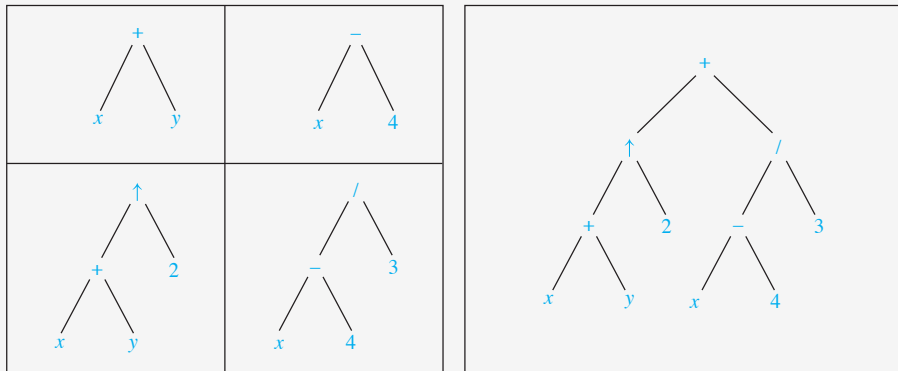
- **Trung thứ tự:** Liệt kê các lá khi đi qua lần đầu, liệt kê đỉnh trong khi đi qua lần thứ 2





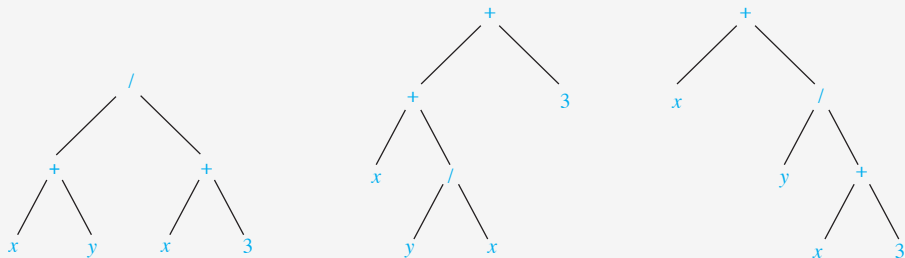
- **Hậu thứ tự:** Liệt kê mỗi đỉnh khi đi qua nó lần trước lần trở về cha của nó.

Cây biểu thức



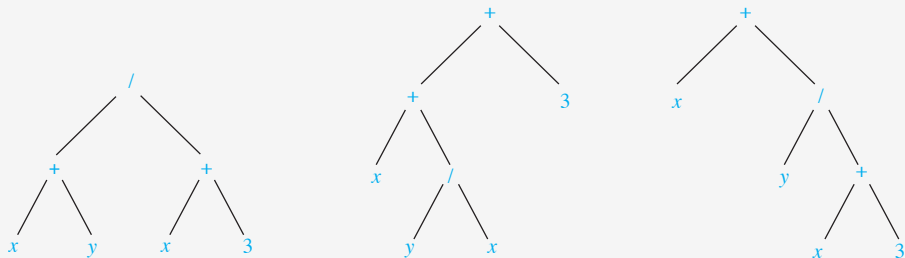
Hình: Cây nhị phân biểu diễn $((x + y) \uparrow 2) + ((x - 4) / 3)$

Tính nhập nhằng của dạng trung tố



Các cây có gốc biểu diễn biểu thức:

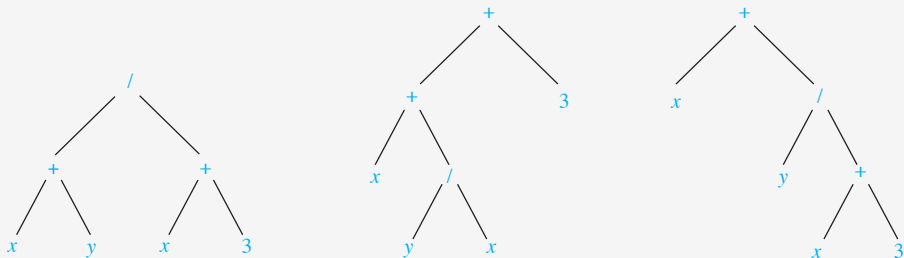
Tính nhập nhằng của dạng trung tố



Các cây có gốc biểu diễn biểu thức:

$$(x + y)/(x + 3),$$

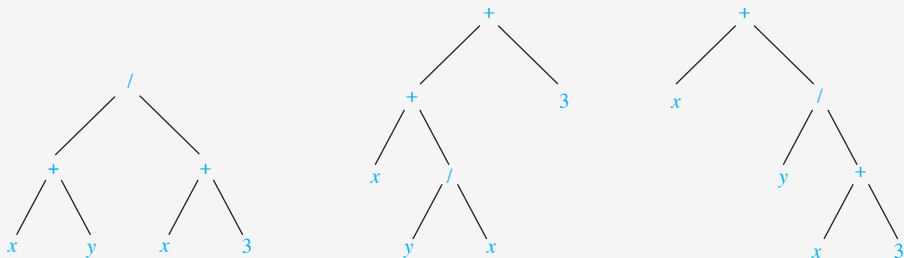
Tính nhập nhằng của dạng trung tố



Các cây có gốc biểu diễn biểu thức:

$$(x + y)/(x + 3), \quad (x + (y/x)) + 3, \quad \text{và}$$

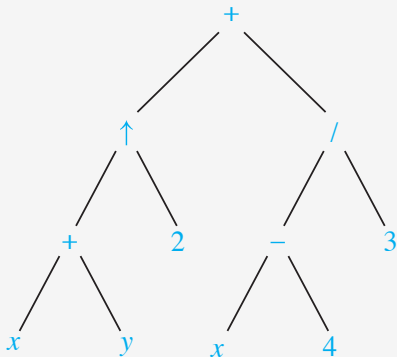
Tính nhập nhằng của dạng trung tố



Các cây có gốc biểu diễn biểu thức:

$$(x + y)/(x + 3), \quad (x + (y/x)) + 3, \quad \text{và} \quad x + (y/(x + 3))$$

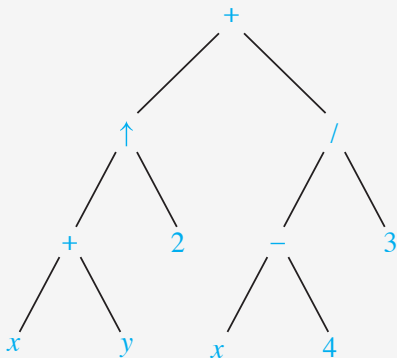
Dạng tiền tố của biểu thức - ký pháp Ba Lan



Dạng tiền tố của biểu thức

$$((x + y) \uparrow 2) + ((x - 4) / 3)$$

Dạng tiền tố của biểu thức - ký pháp Ba Lan



Dạng tiền tố của biểu thức

$$((x + y) \uparrow 2) + ((x - 4) / 3)$$

là

$$+ \uparrow + x y 2 / - x 4 3$$

Tính giá trị của biểu thức tiền tố

$$+ \quad - \quad * \quad 2 \quad 3 \quad 5 \quad / \quad \uparrow \quad 2 \quad 3 \quad 4$$

$$2 \uparrow 3 = 8$$

$$+ \quad - \quad * \quad 2 \quad 3 \quad 5 \quad / \quad 8 \quad 4$$

$$8 / 4 = 2$$

$$+ \quad - \quad * \quad 2 \quad 3 \quad 5 \quad 2$$

$$2 * 3 = 6$$

$$+ \quad - \quad 6 \quad 5 \quad 2$$

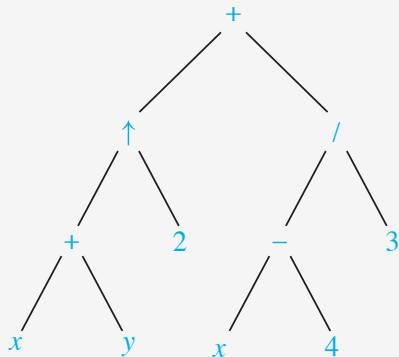
$$6 - 5 = 1$$

$$+ \quad 1 \quad 2$$

$$1 + 2 = 3$$

Value of expression: 3

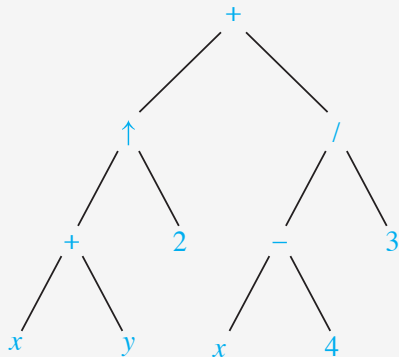
Dạng hậu tố của biểu thức - Ký pháp Ba Lan ngược



Dạng hậu tố của biểu thức

$$((x + y) \uparrow 2) + ((x - 4) / 3)$$

Dạng hậu tố của biểu thức - Ký pháp Ba Lan ngược



Dạng hậu tố của biểu thức

$$((x + y) \uparrow 2) + ((x - 4)/3)$$

là

$$x \ y \ + \ 2 \ \uparrow \ x \ 4 \ - \ 3 \ / \ +$$

Tính giá trị của biểu thức hậu tố

7 2 3 * - 4 ↑ 9 3 / +

└──────────┘

$$2 * 3 = 6$$

7 6 - 4 ↑ 9 3 / +

└──────────┘

$$7 - 6 = 1$$

1 4 ↑ 9 3 / +

└──────────┘

$$1^4 = 1$$

1 9 3 / +

└──────────┘

$$9 / 3 = 3$$

1 3 +

└──────────┘

$$1 + 3 = 4$$

Value of expression: 4

Cây có gốc biểu diễn mệnh đề

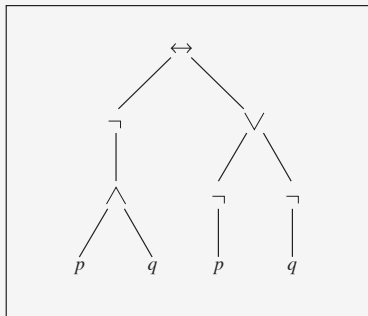
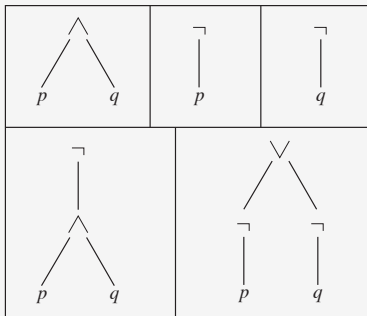
- Tìm cây có gốc biểu diễn cho mệnh đề phức hợp

$$(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$$

Cây có gốc biểu diễn mệnh đề

- Tìm cây có gốc biểu diễn cho mệnh đề phức hợp

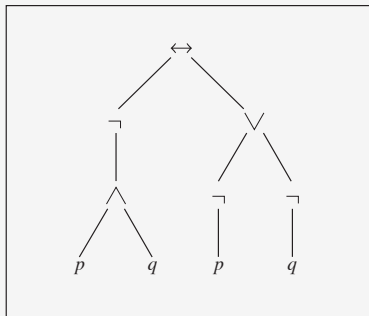
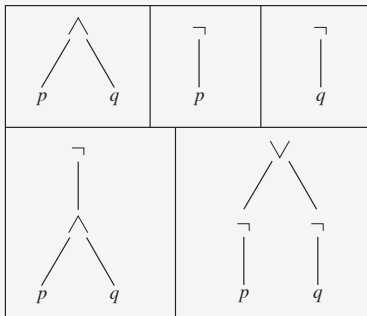
$$(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$$



Cây có gốc biểu diễn mệnh đề

- Tìm cây có gốc biểu diễn cho mệnh đề phức hợp

$$(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$$



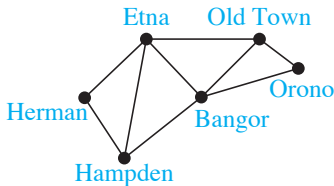
- Dạng tiền tố, trung tố, và hậu tố của biểu thức này là gì?

Nội dung

- 1 Duyệt cây
 - Hệ địa chỉ phổ dụng
 - Các thuật toán duyệt cây
 - Ký pháp trung tố, tiền tố, và hậu tố
- 2 Cây khung
 - Mở đầu
 - Tìm kiếm ưu tiên chiều sâu
 - Tìm kiếm ưu tiên chiều rộng
- 3 Cây khung nhỏ nhất

Bài toán

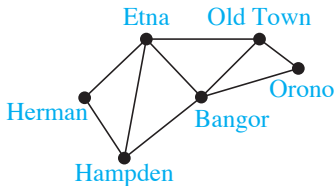
- Hệ thống giao thông của Maine được mô tả như dưới đây.
- Cách duy nhất để đi lại giữa những con đường trong mùa đông là phải cào tuyết thường xuyên.
- Tìm cách cào tuyết một số **ít nhất** các con đường sao cho luôn luôn có đường thông suốt nối hai thành phố bất kỳ.



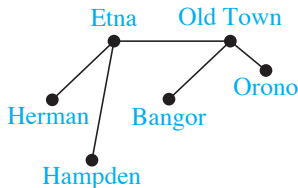
(a)

Bài toán

- Hệ thống giao thông của Maine được mô tả như dưới đây.
- Cách duy nhất để đi lại giữa những con đường trong mùa đông là phải cào tuyết thường xuyên.
- Tìm cách cào tuyết một số **ít nhất** các con đường sao cho luôn luôn có đường thông suốt nối hai thành phố bất kỳ.



(a)

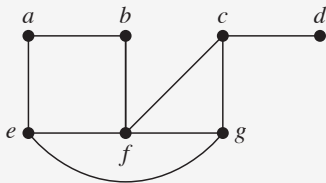


(b)

Cây khung

Định nghĩa

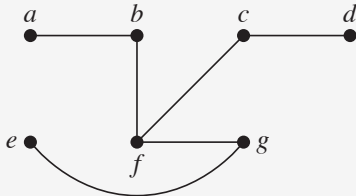
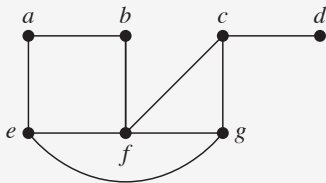
Cho G là một đồ thị đơn. Một cây được gọi là **cây khung** của G nếu nó là một đồ thị con của G và chứa mọi đỉnh của G .



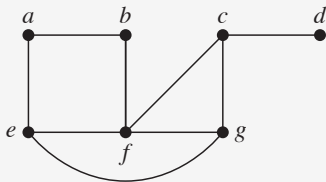
Cây khung

Định nghĩa

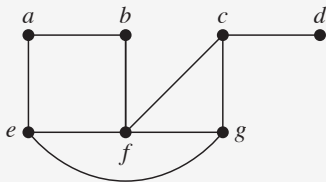
Cho G là một đồ thị đơn. Một cây được gọi là **cây khung** của G nếu nó là một đồ thị con của G và chứa mọi đỉnh của G .



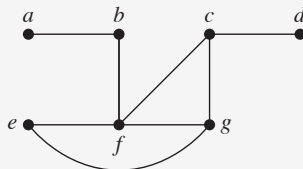
Tìm cây khung bằng cách xóa cạnh



Tìm cây khung bằng cách xóa cạnh

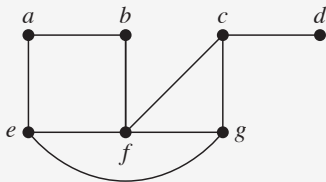


\rightarrow

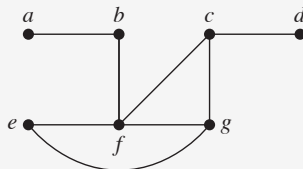


Edge removed: $\{a, e\}$

Tìm cây khung bằng cách xóa cạnh

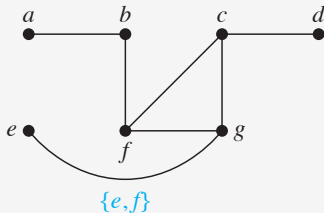


→



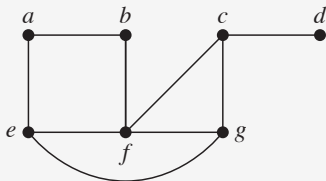
Edge removed: $\{a, e\}$

→

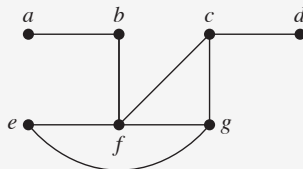


$\{e, f\}$

Tìm cây khung bằng cách xóa cạnh

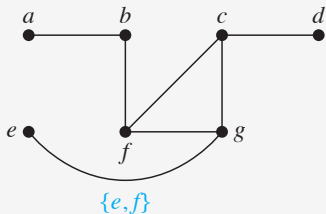


→

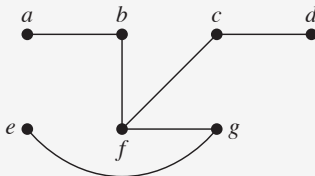


Edge removed: $\{a, e\}$

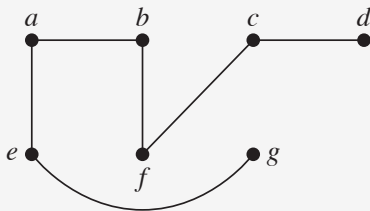
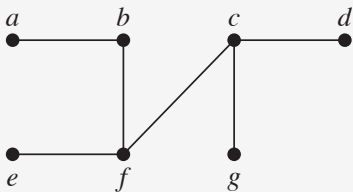
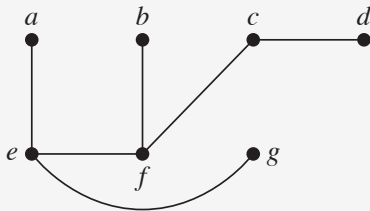
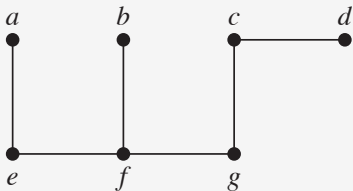
→



→



Một đồ thị có thể có nhiều cây khung



Hình: Một vài cây khung có của đồ thị trước

Định lý

Một đơn đồ thị là liên thông nếu và chỉ nếu nó có cây khung.

Phương pháp tìm cây khung của đồ thị

- Xuất phát từ một đỉnh tùy ý làm gốc
- Xây dựng đường đi từ gốc bằng cách ghép các cạnh vào sao cho mỗi cạnh được ghép sẽ nối một đỉnh trên đường đi với một đỉnh còn chưa thuộc đường đi.

Ứng dụng thuật toán DFS để tìm cây khung

procedure DFS(Đồ thị liên thông G)

T = cây chỉ chứa một đỉnh u của G ;

 visit (u);

procedure visit(Đỉnh v của G)

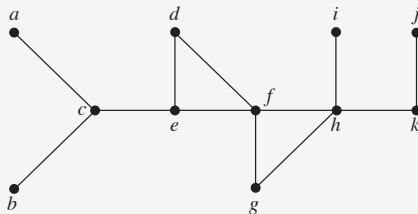
for mỗi đỉnh w liền kề với v và $w \notin T$ **do**

 thêm đỉnh w và cạnh $\{v, w\}$ vào cây T ;

 visit (w);

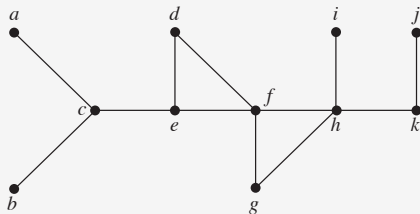
end

Dùng DFS

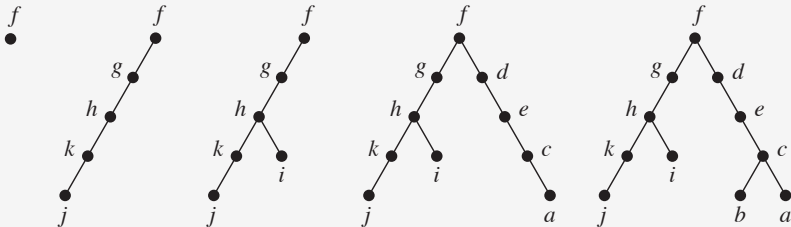


Hãy dùng thuật toán DFS để tìm cây khung cho đồ thị trên.

Dùng DFS



Hãy dùng thuật toán DFS để tìm cây khung cho đồ thị trên.



Ứng dụng thuật toán BFS để tìm cây khung

procedure BFS(Đồ thị liên thông G)

T = cây chỉ chứa một đỉnh u của G

L = danh sách rỗng

 thêm đỉnh u vào danh sách L

while $L \neq \emptyset$ **do**

 xóa đỉnh đầu tiên v ra khỏi danh sách L

for mỗi đỉnh w liên kề với v **do**

if w chưa bao giờ nằm trong L và $w \notin T$ **then**

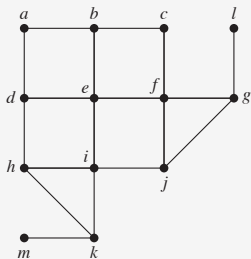
 thêm đỉnh w vào cuối danh sách L

 thêm đỉnh w và cạnh $\{v, w\}$ vào cây T

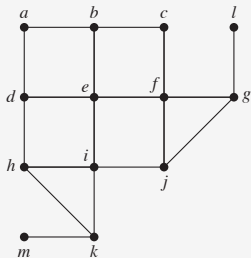
end

end

end

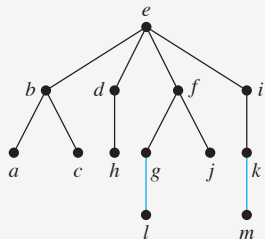
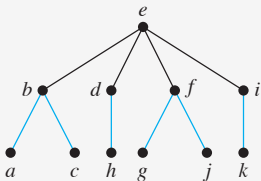
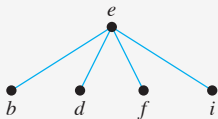


Hãy dùng thuật toán BFS bắt đầu từ e để tìm cây khung cho đồ thị trên.



Hãy dùng thuật toán BFS bắt đầu từ e để tìm cây khung cho đồ thị trên.

e



Nội dung

- 1 Duyệt cây
 - Hệ địa chỉ phổ dụng
 - Các thuật toán duyệt cây
 - Ký pháp trung tố, tiền tố, và hậu tố
- 2 Cây khung
 - Mở đầu
 - Tìm kiếm ưu tiên chiều sâu
 - Tìm kiếm ưu tiên chiều rộng
- 3 Cây khung nhỏ nhất

Cây khung nhỏ nhất

Định nghĩa

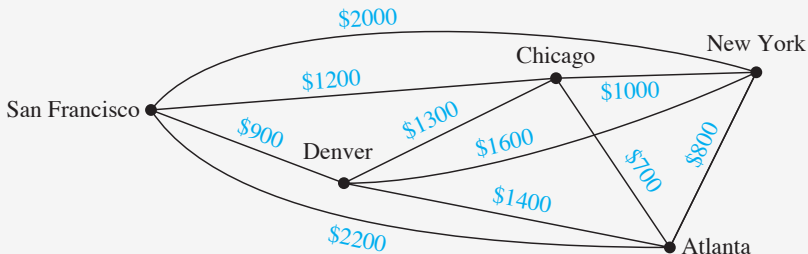
Cây khung nhỏ nhất trong một đồ thị liên thông có trọng số là một cây khung có tổng trọng số trên các cạnh của nó là nhỏ nhất.

Cây khung nhỏ nhất

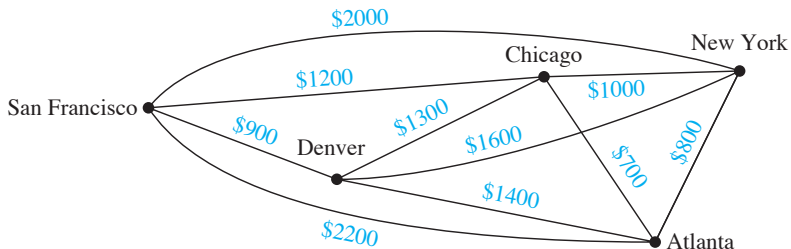
Định nghĩa

Cây khung nhỏ nhất trong một đồ thị liên thông có trọng số là một cây khung có tổng trọng số trên các cạnh của nó là nhỏ nhất.

Hãy tìm cây khung nhỏ nhất của đồ thị sau



Ví dụ



Chọn	Cạnh	Chi phí
1	{Chicago, Atlanta}	700\$
2	{Atlantan, New York}	800\$
3	{Chicago, San Francisco}	1200\$
4	{San Francisco, Denver}	800\$
Tổng chi phí:		3600\$

Thuật toán Prim-Jarník

function Prim(Đồ thị G liên thông có trọng số với n đỉnh)

T = cạnh có trọng số nhỏ nhất

for $i = 1$ **to** $n - 2$ **do**

e = cạnh có trọng số nhỏ nhất trong G thỏa mãn: e liên thuộc với một đỉnh thuộc T và $T \cup \{e\}$ không có chu trình

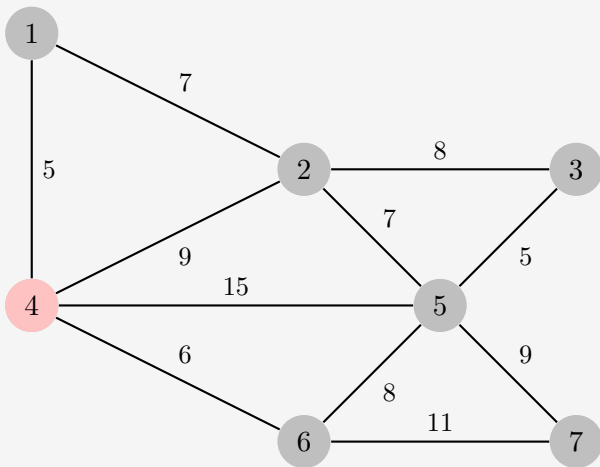
$T = T \cup \{e\}$

end

return T

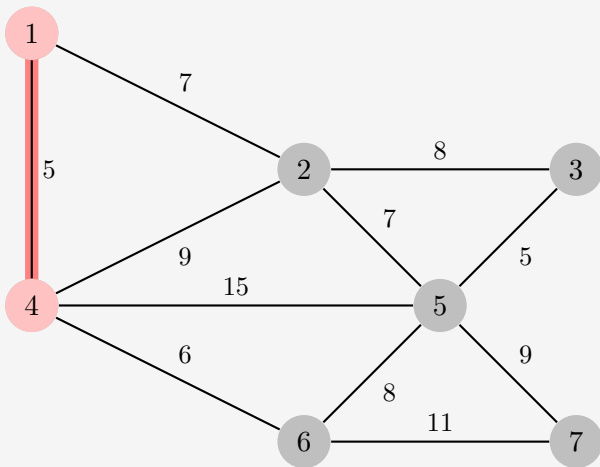
/* T là cây khung nhỏ nhất */

Ví dụ: Thuật toán Prim-Jarník



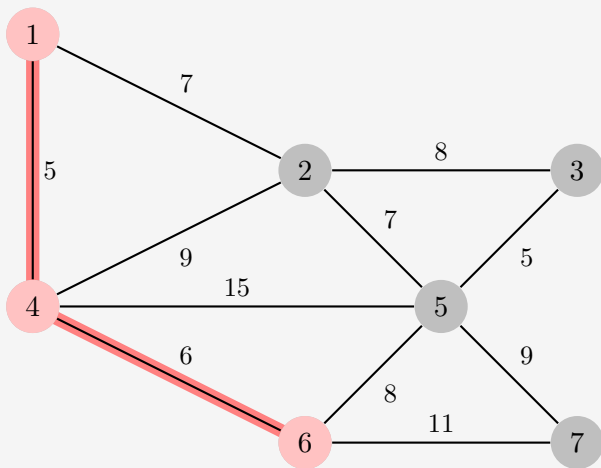
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník



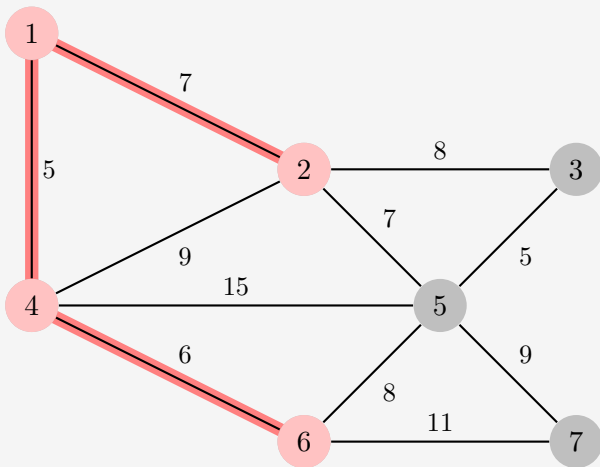
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník



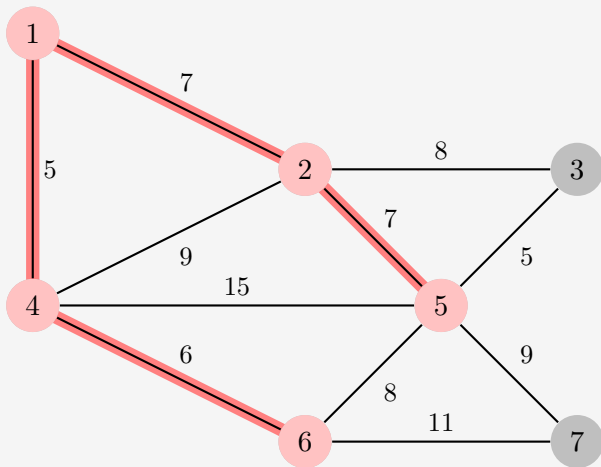
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník



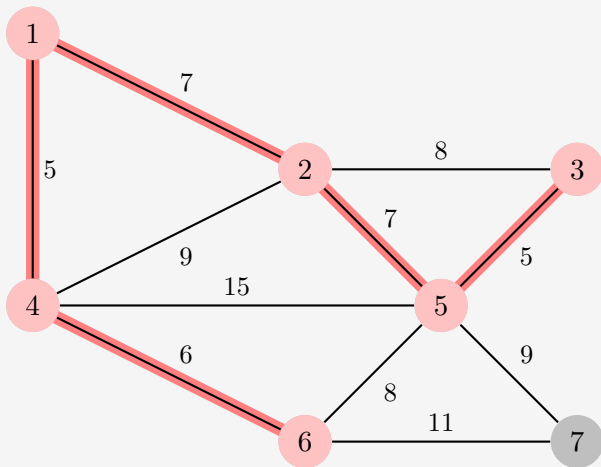
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník



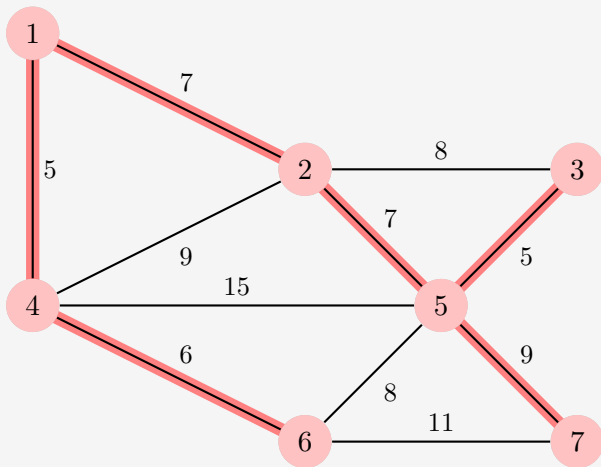
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník



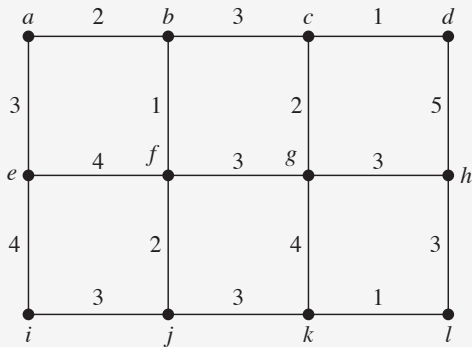
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník

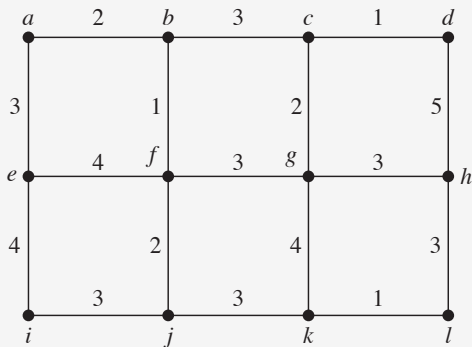


Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Prim-Jarník



Ví dụ: Thuật toán Prim-Jarník

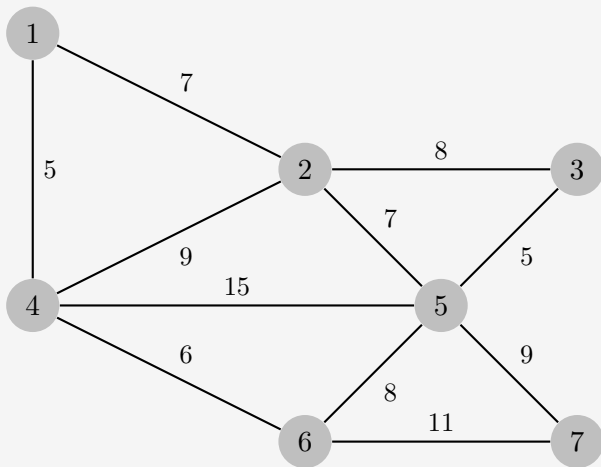


Bước	Cạnh	Trọng số
1	$\{b, f\}$	1
2	$\{a, b\}$	2
3	$\{f, j\}$	2
4	$\{a, e\}$	3
5	$\{i, j\}$	3
6	$\{f, g\}$	3
7	$\{c, g\}$	2
8	$\{c, d\}$	1
9	$\{g, h\}$	3
10	$\{h, l\}$	3
11	$\{k, l\}$	1
Tổng:		24

Thuật toán Kruskal

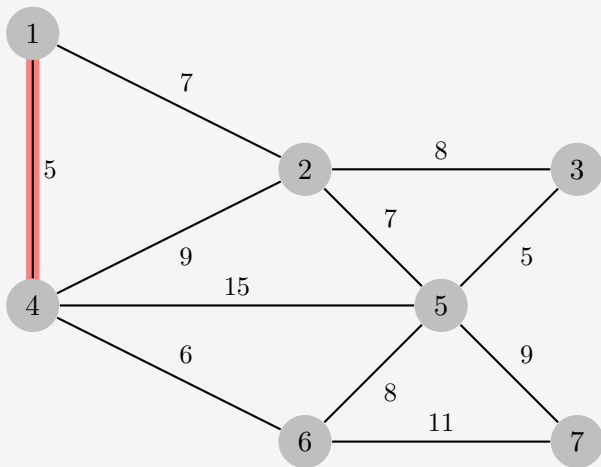
```
function Kruskal(Đồ thị  $G$  liên thông có trọng số với  $n$  đỉnh)
     $T =$  đồ thị rỗng
    for  $i = 1$  to  $n - 1$  do
         $e =$  cạnh trong  $G$  với trọng số nhỏ nhất thỏa mãn đồ thị  $T \cup \{e\}$ 
        không có chu trình
         $T = T \cup \{e\}$ 
    end
    return  $T$                                 /*  $T$  là cây khung nhỏ nhất */
```

Ví dụ: Thuật toán Kruskal



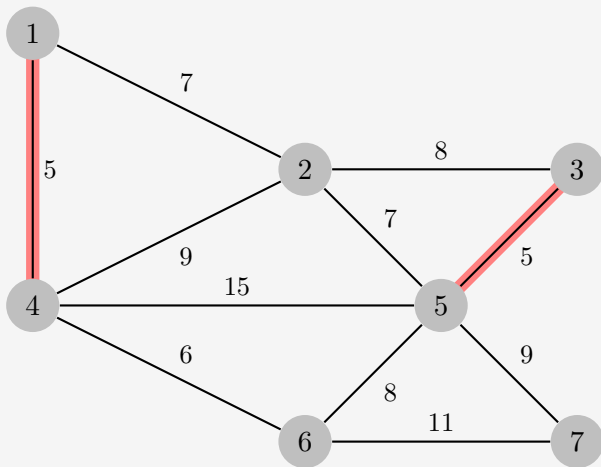
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal



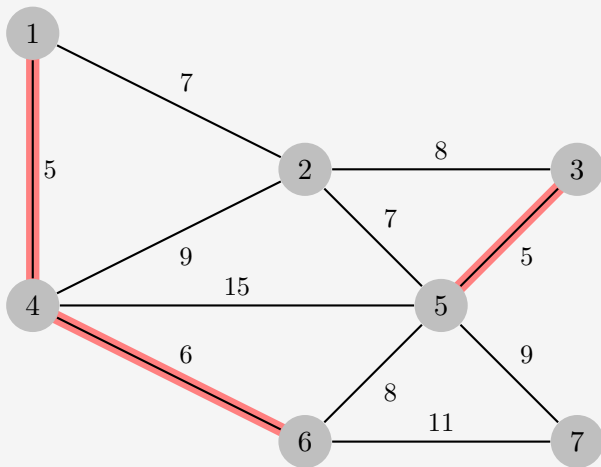
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal



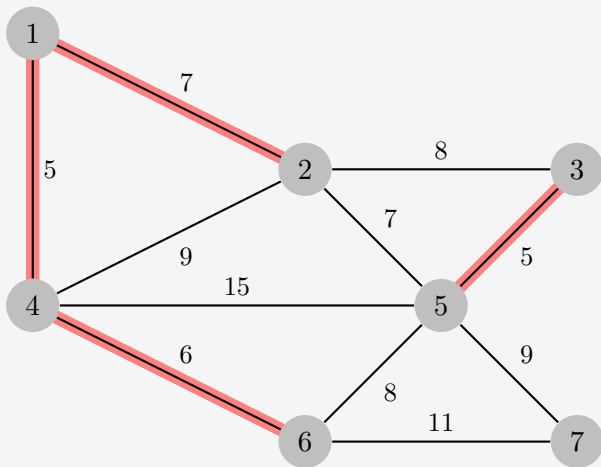
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal



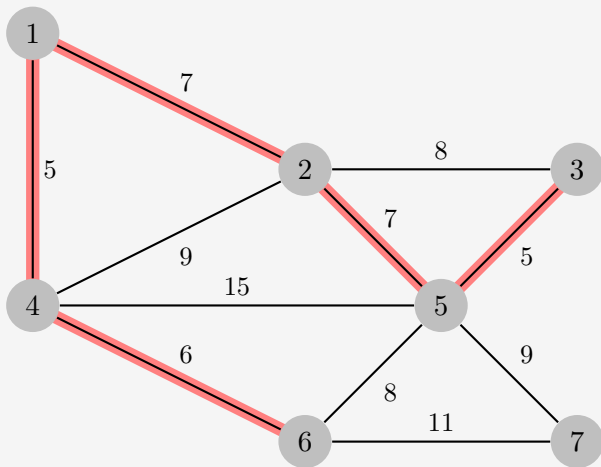
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal



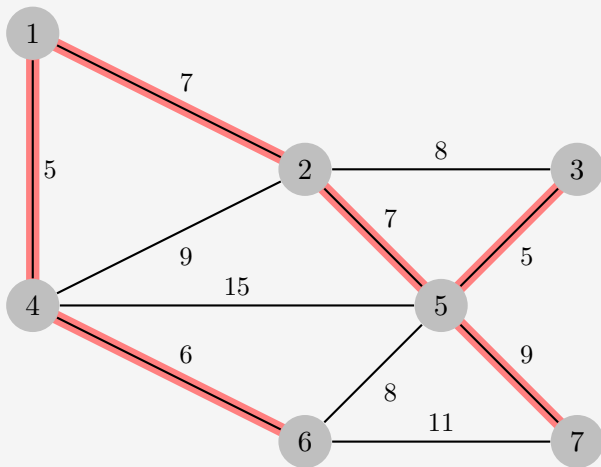
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal



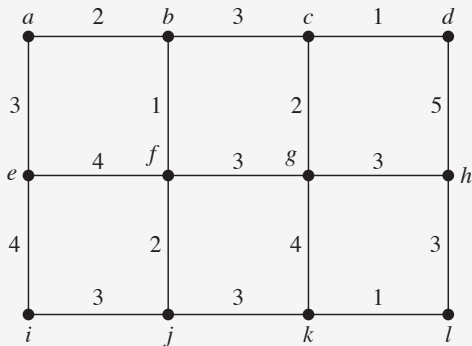
Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal

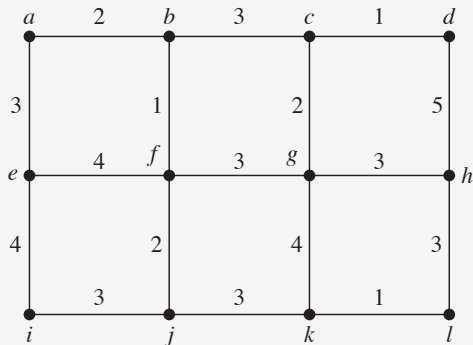


Hình: Nguồn: tikz examples

Ví dụ: Thuật toán Kruskal



Ví dụ: Thuật toán Kruskal



Bước	Cạnh	Trọng số
1	$\{c, d\}$	1
2	$\{k, l\}$	2
3	$\{b, f\}$	2
4	$\{c, g\}$	3
5	$\{a, b\}$	3
6	$\{f, j\}$	3
7	$\{b, c\}$	2
8	$\{j, k\}$	1
9	$\{g, h\}$	3
10	$\{i, j\}$	3
11	$\{a, e\}$	1
Tổng:		24